izdaje BIGZ · izlazi jednaput mesečno

računari

specijalno izdanje časopisa „Galaksija"
jul 1987
cena 500 din

28



ekskluzivno
„spektrum 3 plus"
sa tri plusa

„komodor 64"
pregršt novih igara

svakodnevni algoritmi

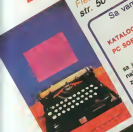
umetak na 32 strane
sve ms dos funkcije

„amstrad"
krug za pet stotinki

„atari st"
štamparija na stolu

U ovom broju

Komercijalni softver:
Fleet Street Publisher
str. 50



Sa vama i u avgustu

KATALOG PC SOFTVERA

sa jezgrovitim opisom najkapitalnijih programa za IBM PC XT/AT i kompatibilnu bratiju iz pera Arifa Agovića

i

RAZBARUŠENI SPRAJTOVI

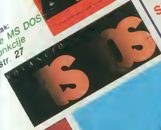
na povećanom broju strana.



Snimio Vladimir Simović

Bazen hotela Jugoslavija

Sledeći, avgustovski broj izlazi iz štampe krajem jula



otak:
ve MS DOS
unkcije
str. 27



Nova HP generacija:
RISC u Beogradu
str. 4



Spektrum 3 plus:
„Duga" sa tri plusa
str. 9

28

računari

Izdaje
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar vojvode Mišića 17

Telefoni
650-181 (redakcija)
653-748
650-528 (prodaja)
651-793 (propaganda)

Generalni direktor
Dobrosav Petrović

V d direktor sektora izdavačko-novinska delatnost
Antun Martić

V d glavni i odgovorni urednik
Gavrilo Vučković

Odgovorni urednik
Jova Regasek

Likovno-grafičko uređenje
Mirko Popov

Redakcija časopisa „Galaksija
Tanasije Gavranović urednik
Esad Jakupović v d zamenik glavnog i odgovornog urednika
Aleksandar Milinković urednik
Jova Regasek odgovorni urednik
Zorka Simović sekretar redakcije
Srđan Stojančev novinar
Gavrilo Vučković v d glavni i odgovorni urednik

Stručni saradnici
Radomir A Mihajlović Dejan Ristenović Dušan Slavić Nevenka Spalević Anđelko Zgorelec

Spoljna redakcija
Branko Đaković Dejan Ristanović Jelena Rupnik Jovan Skuljan prof dr Dušan Slavić Nevenka Spalević Vlada Stojiljković Zoran Životić

Stalni saradnici
Neda Aleksić Ninoslav Čabrić Branko Đaković Voja Gašić Đenko Hebreng Željko Jurić Radomir A Mihajlović Zvonimir Makovec Blažimir Miše Dejan Muhamedagić Ivan Nador Zoran Obradović Miodrag Potkonjak Dejan Ristanović Jelena Rupnik Dušan Slavić Jovan Skuljan Nevenka Spalević Zvonimir Vistrička Žarko Vukosavljević Anđelko Zgorelac Zoran Životić

Izdavački savet „Galaksije
Dr Rudi Debijadi prof dr Branislav Dimitrijević (predsednik) Radovan Drašković Tanasija Gavranović Živorad Gilšić Esad Jakupović, Velizar Maslać Nikola Pajić Željko Parunović prof dr Momčilo Ristić Vlada Ristić dr inž Milorad Teofilović Vidojko Veličković Velimir Vasović Milivoje Vuković

Štampa
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar vojvode Mišića 17

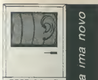
Adresa redakcije
11000 Beograd
Bulevar vojvode Mišića 17/III

Rukopisi se ne vraćaju
PRETPLATA ZA RAČUNARE
za Jugoslaviju
RO BIGZ 60802-603-23264
za jednu godinu — 6 000
za pola godine — 3 000
Za inostranstvo
RO BIGZ 60811-620-16-101-820701-598-03377
22 US$ 39 DM 13 fstg 33 Švfr 136 Švkr 131 Ffr lit 12 000 din
Na osnovu mišljenja Republičkog sekretarijata za kulturu broj 413-77/72-03 i „Službenog glasnika broj 26/72 ovo izdanje oslobođeno je poreza na promet

Na tradicionalnom majskom Sajmu tehnike u Beogradu bilo je više računara nego ikada dosada. Po prvi put, međutim, domaći izlagači su počeli da primenjuju svetske manire. Tako je štand Ei-Hanivela podsećao na štandove giganata računarstva na svetskim izložbama. Na svom štandu fotoreporter Vladimir Simović sreo je Snežanu Međedović, manekenku iz Beograda, i nije se mnogo premišljao da ovekoveči ovaj prizor. Mala pomoć organizatora Branislava Đorđevića i snimak je bio gotov



Sadržaj





Novi PS/2 časopisi

Velike računare ne čine samo hardver i sistemski softver — za uspeh je neophodna i ogromna podrška nezavisnih firmi i časopisa. Ako se po jutru dan poznaje, IBM-ova nova serija PS/2 da bi bila izvanredno podržana — još se nije osušila štamparska boja na prospektima a pojavila su se dva specijalizovana časopisa. PS User je dopuna popularnog časopisa Which PC i bavi se pratežno karakteristikama nove serije računara i njenih proširenja dok je Power PC User namenjen malo naprednijim korisnicima. Kolegama iz novih britanskih časopisa želimo mnogo dobrih tekstova koje ćemo vrlo rado koristiti!

Superbrzi „sitizen"

Poznata firma Citizen je svoju porodicu štampača dopunila modelima MSP 50 i MSP 55. Radi se o devetopinskim štampačima sa kvalitetnim otiskom koji su opremljeni jednim specijalitetom: kartridžima za dodavanje novih setova znakova. Uz štampač dobijate kartridž sa NLQ italikom, a za dvadesetak funti možete da kupite EPROM sa nekom drugom azbukom. Štampači su izuzetno brzi: 300 običnih ili 50 NLQ znakova u sekundi.

D. R.

RISK u Beogradu

U vreme najveće žurbe oko priprema za sajam tehnike, u Beogradu se tiho, daleko od očiju radoznalaca, pripremao jedan drugi, za našu računarsku scenu izuzetno važan događaj — prezentacija novih proizvoda Hewlett-Packarda uključujući i HP 9000 Model 840 — krunsku mašinu nove RISC generacije.

Prezentacija je počela 26. maja u tišini Sava Centra, daleko od sajamske buke i nervoze. Pred očima naših tehničkih profesionalaca našla su se najnovija dostignuća firme koja je u prošloj godini imala najveći porast vrednosti akcija (preko 47%) od svih američkih firmi visoke računarske tehnologije.

Čitaoci „Računara" su već upoznati sa novom Packardovom generacijom RISC procesora (HP-Precision Architecture, u daljem tekstu HP-PA) koji su na ovoj prezentaciji imali zapaženo mesto. Beograd se uvrstio u red nekoliko evropskih metropola u kojima je predstavljen HP-9000 Model 840 (HP-840 u daljem tekstu), do sada najmoćniji izdanak HP-PA filosofije namenjen naučnom i inženjerskom radu. Ovaj računar je, takođe, prvi iz nove generacije HP mašina koji radi pod UNIX operativnim sistemom, što će reći da u startu raspolaže ogromnom količinom softvera, uz novi Packardov kompajlerski sistem koji su čitaoci „Računara" takođe upoznali (RA21/32), moćnu bazu podataka i podršku vezivanju u mrežu.

Ovakvim potezom HP je dao najveći mogući podsticaj razvoju RISC mašina jer je u praksi dokazao da su one čak i pod vrlo standardizovanim operativnim sistemom pravi brzinski šampioni. HP-840 je brži od nadaleko čuvenog VAX-780 „svega" 8,2 puta i „tuče" sve VAX-ove do modela 8600 uz znatno nižu cenu. Ono što je za sve nas prisutne na prezentaciji bilo možda i važnije od ovih cifara je pojava radne stanice 825, koja je dvostruko sporija od 840 (što će reći oko 4 puta brža od VAX-780) ali se zato cenom spušta u područje konvencionalnih radnih stanica. Pojava tog modela, modela 850 (koju „tuče" i „mainframe" mašine) kao i najava HP-PA procesora na čipu svedoči o velikoj pobedi RICS filosofije, čijim će „izdancima" „Računari" i u buduće posvećivati dužnu pažnju.

Ako je HP danas vodeći u RICS području, to još ne znači da zapostavlja „konvencionalne" tehnologije. Dokaz toga su radne stanice 330 i 350 koje rade sa 68020 procesorima na 16,7 odnosno 25 MHz, pod UNIX-om i sa specijalizovanim grafičkim čipovima čije rezultate vidite na slikama. U Sava Centru smo videli dve stanice opremljene Packardovim CAD-paketima i grafičkim monitorima rezolucije 1280×1024 sa 16 bitnih ravni (može se raditi i sa 24 odnosno 32). Stanice su bile povezane u brzu lokalnu mrežu sa diskom od 132 Mb i kolor ploterom koji omogućava projektovanje osnovnog crteža na jednoj stanici, prenos na drugu radi 3D-dorade, vraćanje na prvu radi izrade dokumentacije i izdavanje crteža na ploteru — sve za desetak minuta. Sve što se događalo na monitorima moglo se videti i na platnu — zahvaljujući video projektoru. Videli smo i jednu „sporednu" primenu ovih stanica — animaciju u realnom vremenu, omogućenu izuzetno brzim specijalizovanim čipovima. Kada se na sve ovo doda vrhunski HP-dizajn i „prijateljsko" ponašanje programa (rad sa prozorima i menijima je ovde toliko brz da je njihovo otvaranje/zatvaranje za korisnika trenutno) dobija se sistem sa ubedljivo najboljim odnosom cena/kvalitet.

Iako smo od ranije znali da HP gazi krupnim koracima u području RISC tehnologija i CAD sistema, HP-Artificial Intelligence Development System nas je prilično iznenadio. Pre godinu dana je, doduše, objavljeno da se „nešto radi", ali bez nagoveštaja da će to „nešto" biti uskoro na tržištu. Iznenađenje je tim veće što veštačka inteligencija ima za sada uglavnom akademsku i istraživačku, a ne razvojnu i proizvodnu primenu. Kako to uskladiti sa renomeom Hewlett-Packarda kao proizvođača mernih uređaja koji računara smatra, pre svega, proizvodnim alatom? O daljoj sudbini veštačke inteligencije u Packardu se mudro ćuti, ali nije daleko od pameti da se uskoro mogu pojaviti „veštački inteligentni" merni sistemi, utoliko pre što Common LISP, Q-PROLOG i Z-PROLOG kompajleri, koji čine razvojni sistem, imaju mogućnost povezivanja sa C, fortran i mašinskim programima, što omogućava laku dogradnju upravljačkih modula.

Sve u svemu, Hewlett Packardove prezentacija je ostavila utisak brižljivo planiranog i dobro ukomponovanog nastupa na nivou koji nije čest u našim prostorima. O odjeku koji je ovaj nastup imao među ozbiljnim kupcima ne treba posebno govoriti. Puna sala sa oko 250 ljudi i „nedostatak mesta" u sali za demonstracije dovoljno govore sami za sebe.

Žarko Berberski



HP na Elektrotehničkom fakultetu

Poznata američka firma Hewlett Packard je početkom aprila poklonila beogradskom Elektrotehničkom fakultetu razvojni sistem HP 64000 UX MDE vredan 70 000 dolara! Radi se o jednom od najnovijih Packardovih računara koji se koristi za projektovanje mikroprocesorske opreme zasnovane na čipovima raznih proizvođača Motorolini 68000 68020 i 68030, intelovi 8086, 80186 i 80286 Nationalov 32332 i drugi — sistem bi mogao da se koristi i za razvoj opreme zasnovane na Intelovom 80386 ali potreban softver još nije pristigao u Beograd

Radne stanice HP 64000 UX MDE se sastoji od centralne jedinice sa hard diskom, semi grafičkog terminala, dve mini disk jedinice po 3 5 inča i think-jet štampača Što se softvera tiče, u osnovnu konfiguraciju spadaju kros kompajleri za paskal i C i odgovarajući debageri

Ova donacija je još jedan dokaz dalekovidosti firme Hewlett Packard koja je samo u toku prošle godine raznim (najčešće obrazovnim) organizacijama poklonila svoje proizvode vredne preko 60 miliona dolara. Radne stanice HP 64000 već uspešno koristi nekoliko jugoslovenskih tehničkih fakulteta

D R

Acorn u zamahu

Posle dugogodišnje krize Acorn pokazuje solidne poslovne rezultate — polugodišnji obrt od 46 7 miliona funti je za čak 1 4 miliona funti veći od planiranog Veliki deo ovog profita potiče od prodaje računara iz serije „master" van Velike Britanije — „master" se prodaje u Indiji, Francuskoj i Italiji, a na pomolu je i sporazum sa Kinezima S druge strane, čuli smo da su u Jugoslaviji prodata čitava dva „master" računara!

D R

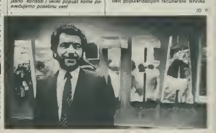
PC sa popustom

Amstrad je najavio veliki letnji popust — za sve Amstrad PC računare kupljene između 1 juna i 31 avgusta cena će biti snižena za čitavih 150 funti Tako PC 1512 sa 512 kilobajta RAM-a, dvostrukom flopi disk jedinicom monitorom štampačem i tekst procesorom košta samo 700 funti+VAT (normalna cena 850 funti) Požurite u Englesku!

D R

Amstrad protiv IBM-a

I pored izvesnih kritika Amstrad PC se izvanredno prodaje — saznajemo da Šugar u Britaniji trenutno prodaje više PC računara nego sam IBM! Prodaji će, jasno koristiti i veliki popust kome posvećujemo posebnu vest



Savez za informatiku

Osamnaestog aprila je u Beogradu održana Osnivačka konferencija „Saveza organizacija za informatiku i računarsku tehniku Jugoslavije", tako su odgovarajući republički savezi Srbije Hrvatske Makedonije i Vojvodine dobili telo koje će koordinirati njihov rad

Osnovni zadatak novoformiranog Saveza je da radi na ostvarivanju ciljeva udruženih republičkih i pokrajinskih organizacija i na zajedničkim aktivnostima u afirmaciji širenju i razvijanju tehničke kulture u oblasti informatike i računarske tehnike kod dece omladine radnih ljudi i građana. Savez će osim toga afirmisati i podsticati stvaralaštvo u oblasti informatike, sarađujući pri tom sa raznim naučnim i obrazovnim institucijama u zemlji i svetu

Na Osnivačkoj konferenciji Saveza usvojena je programska orijentacija i program rada za 1987 godinu Program rada, jasno započinje organizacionim učvršćivanjem Saveza, izradom normativnih akata i registrovanjem kod nadležnih organa uprava Planiraju se i savezne manifestacije, ostvarivanje međunarodne saradnje, štampanje jedinstvenog adresara svih društvenih organizacija (klubovi, sekcije, savezi) koje se bave računarskom tehnikom i izdavanje informativnog biltena

Savez će kod nadležnih organa biti registrovan za trinaest društvenih delatnosti među koje spadaju istraživačko-razvojne usluge u društvenim delatnostima usluge reklame i ekonomske propaganda, izrada tehničke dokumentacije knjigovodstvene usluge osnovno i ostalo obrazovanje izdavačku, novinsko-izdavačku kulturno-propagandnu i pomoćnu naučno-istraživačku delatnost.

Poželimo , Savezu Organizacija za informatiku i računarsku tehniku Jugoslavije uspešnu konsolidaciju i plodan rad „Računari" će se potruditi da obavesta čitaoce o svim akcijama koje će voditi ova prva opštejugoslovenska organizacija koja se bavi popularizacijom računarske tehnike

(D R

Automatizacija poslovanja

Baze podataka i njihove zamke

(Ili: svi imaju, samo mi namamo bazu podataka)

Više ne možete da izdržite da sa suočavate sa strašnom letinom: sva organizacije vašeg ranga imaju bazu podataka, čak i mnoge čiji je značaj neuporedivo manji od vašeg, jedino još vi precrtavate spiskove mastiljavom olovkom. Krajnje je vreme da i vi nešto preduzmete. Vaše odluka je za svaku pohvalu. Samo, kako ući u avanturu zvanu „kompjuterizacija poslovanja" a ipak izvući živu glavu

Instituti, biblioteka pa i veće firme koja nabavljaju stručnu literaturu, nastoje da obezbeda brži pristup saradnika koji imaju zadatak da pročitaju časopise (svaki iz svoje uže specijalizacija) i da interesantne članke obrada u obliku kratke anotacije. Ove anotacije se, zatim, mogu čuvati u obliku kartoteke ili publikovati u nekakvom biltenu. I u jednom i u drugom slučaju anotacije treba da budu klasifikovane tako da se do njih može relativno lako doći. U tu svrhu se najčešće koristi univerzalna decimalne klasifikacija (UDK)

Vi o tome sve znate. Znete i da je potrebno otvoriti nekoliko kartoteka sa anotacijama po naslovu, autoru ili vrsti informacija. Kako stručni članci uglavnom pripadaju ne samo jednoj vrsti, treba ih rasporediti u nekoliko grupa, što dovodi do umnožavanja kataloških listića. Kataloški listić mora da bude na kartonu koji se ne može propustiti kroz mašinu za foto-kopiranje, što znači da se moraju prekucavati pisaćom mašinom. Ovo sa sobom povlači priličan broj grešaka. Opterećenost daktilografa drugim poslovima dovodi do kašnjenja u prekucavanju, a ovo pak do toga da informacija najčešće buda već uveliko zastarela kad uđe u katalog. Svemu ovoma doprinose i kašnjanja isporuke stručnih časopisa, dugo zadržavanja časopisa kod stručnih saradnika itd. Informacija najčešće kasni oko godinu dana, kad postaje pristupačna korisniku

Vaša odluka je već zrela: nabavićete kompjuter i stvoriti svoju bazu podataka

Prva zabluda

Ubrzaće se posao i pristup informacijama

Neće, jer će isporuka časopisa i dalje kasniti, saradnici će i dalje da ih zadržavaju, a operateri će imati i mnoge druge poslove, pa će ukucavanja u memoriju ići isto tako sporo. Ali ste bar oslobodili daktilografkinju mukotrpnog prekucavanja

No vas ništa ne može zaustaviti. Pažljivo ćete pročitati sve članke u kojima se govori protiv ovakve automatizacija posla i kupićete računar

Druga zabluda

Poslovanja će se širiti, pa je najbolje odmah uzeti veliki računar

Velika mašine su uvek prevelika za vas. Kupujte samo jeftine mašine. Ako se ispostavi da ste u stanju na njima da radite, na velikima ćete uživati, a za malu će se uvek naći posla. Ako ne možete da radite na maloj mašini, ni velika vam neće pomoći. Bar niste bacili mnogo para, a računar koji ste kupili možete uvek koristiti kao pisaću mašinu. Ali ne preterujte ni u štednji. Nemojte kupiti „komodor 64" ili, ne daj bože, na nekoj carinskoj licitaciji jeftino TRS 80 (pre 10 godina to je bio hit računar)

U redu, kupićete računar. Naravno, pitate se, koji? Vi se u to očito ne razumete, pa ćete pitati nekoga ko to bolje zna. Od 90 ljudi koje budete pitali čućete 80 različitih odgovora. Onih deset će vam verovatno reći da je IBM ipak IBM i zato — neka buda IBM

Treća zabluda

Treba kupiti računar koji najviše preporučuju

Kada vas neko savetuje da kupite neki računar, obično je to isti računar na koma dotični već radi. IBM preporučuju zbog softvera: ima nekih 12 000 programa za tu mašinu. Šteta je samo što su ti programi pravljeni za Amerikanca i za vašu bazu podataka su praktično neupotrebljivi

Za novac koji uložite u IBM PC od 512 KB možete kupiti tri računara „atari 2080ST" sa četiri puta većom memorijom. Pouzdanost u redu im je ista, ali je brzina na strani „starija"

Naravno, vi ste kupili neki treći računar. Bili su ljudi iz servisa, sve su povezali, sve lepo radi. Imate i prekrivač za njega. Lepo ste ga pokrili da ne pada prašina. Niko nije stručan da ga pokrene, i vi ne date nikoma da ga pipne. Da ga ne pokvari!

Četvrta zabluda

Računar treba čuvati od onih koji vole da kuckaju

Nikada niko nije uspeo da pokvari računar radeći sa tastaturom. Činjenica je da je računar pouzdane mašine. Može se dodu-še, dogoditi, da u prvih nekoliko meseci otkaže neki čip. Kad prođu te dečje bolesti obično nastupa period dugotrajne eksploatacije bez većih kvarova

Uz računar ste dobili i nešto programa. Ni jedan od njih ne može da odštampe referativni bilten vaših anotacija. Poslali ste nekoliko ljudi na kurs. Naučili su da uključe računar, a poneko uma i da napiše sledeći program:

```
10 INPUT „Otkucaj svoje ime", A
20 PRINT „ZDRAVO", A
30 END
```

Uviđate da je vrag odneo šalu i da ćete morati da otvorite novo radno mesto za programera

Peta zabluda

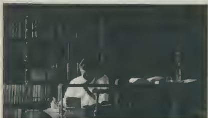
Programera treba primiti u stalni radni odnos

Obim poslova vaše organizacije je ipak ograničen. Dobar programer će to da automatizuje za dve do tri godine. I šta posle sa njim? Primili ste ga u stalni radni odnos a to znači za narednih 40 godina. Kada jednom uradi sve programe, počeće da vam smeta što ne radi. Daćete mu da kuca bilo šta, i on će da se naljuti i ode. A onda ste u situaciji da vam i najbezazlenija greška potpuno zaustavi posao. Jer samo je On u mogućnosti da grešku otkloni. Šta da radite? Da uzmete drugog programera? Taj će sve morati da radi ispočetka. Da napravite ugovor o delu? Biće to veoma skupo, ali jeftinije od prethodne varijante

Šesta zabluda

Dobar programer je samo onaj programer koji ima punovažnu diplomu

Ima i druga varijanta. Primili ste programera sa diplomom kao vrata. On o vašem načinu rada ne zna baš ništa. Prvi put je čuo za UDK. U školi je učio organizaciju podataka na disku iz knjiga koje su pisane još 1960. Za stručne časopise (do danas je bio bez posla) nije imao para. Možda mu je tatka za rođendan poklonio „spektrum". Naravno, učio je Kobol, a o operativnom sistemu vaše mašine nema pojma. Osećanje za logičnost je glavni element onoga



što se zove programerski talent. Kada to osećanje dovede na tako visok stepen da nelogičnosti počne da oseća svuda oko sebe, onda će nelogičnosti toliko da mu smetaju da će se potpuno okrenuti računaru. U narednih nekoliko godina će steći potrebno iskustvo i izgubiti sve prijatelje. Žena će ga napustiti još prve godine i treba da ga napusti.

Pitajte, dakle, kandidate za programere jesu li — razvedeni.

Sedma zabluda

Izmene radi poboljšanja programa ne mogu biti štetne

Želite bazu podataka iz oblasti medicine, građevine, hemije... Problem je uvek isti. Podatke treba uneti u računar, sortirati, ispisati u nekakvoj formi koja bi bila lepa za oko. Ko hoće može ih i pretraživati po nekakvim ključevima.

Postavićemo, zato, ovakav zadatak: hoćemo da napravimo kuvar. U njemu bi bili svi oni recepti kojih, inače, nema u drugim kuvarima. Kad napravimo lepu zbirku od oko 10 000 recepata, odštampaćemo knjigu kakvu svet još nije video. Na kraju tog kuvara imaćemo sledeće indekse.

Indeks autora pojedinih recepata. Ako nam se recept nekog autora mnogo svidi, nećemo valjda prelistavati celi kuvar da bismo otkrili gde su još smešteni gastronomski biseri spomenutog majstora.

Permutirani indeks pojedinih sastojaka. Recimo, želimo da napravimo neko jelo od karfiola. Kupili smo ga još prošle nedelje, već je počeo da se ruši u frižideru. Okrenemo indeks, pronađemo pod karfiol i čitamo: karfiol, jaja, slanina... e, to nećemo, nemamo slanine u kući. Čitamo dalje: karfiol, mleko, luk, limun... nemamo ni limuna... i tako redom. Kad je indeks permutiran, onda svaka odrednica (sastojak recepta) mora jedanput da dođe na prvo mesto. Blizu je pameti da će na slovo, odnosno tamo gde počinje odrednica so, biti najveći spisak, jer sva slana jela sadrže so.

Potrošili smo mnogo hartije da bismo to uvideli, pa ćemo reći da u permutaciju ne smeju da uđu začini. Tu se naravno odmah otvara novi problem. Šta su to začini? Da li je, recimo, celer začin? Naravno da jeste. A pohovani celer? Nije začin, to je glavni sastojak jela. Odluka „izbaciti sve začine" mora da se preinači u „izbaciti so, biber, brašno, ulje i šećer".

Jeste li sigurni da opet nije otvoren neki novi problem?

Osma zabluda

Najbolje je normirati obaveze programera; pošto niko ne može da zna da li on zaista radi ili zabušava, problem se rešava kontrolom izvršavanja norma

Grdno ćete se prevariti ako pokušate da normirate rad vašeg programera (normiranje je danas popularno), pa mu, recimo, odredite normu od 200 programskih linija dnevno. Ne možete ni obrnuto da ga procenjujete na osnovu toga koliko je malo programskih linija uneo u računar. Čak ni na stanovište efikasnosti programa ne možete da mu kažete ni reč. Jer svi programi rade. Sporo? Ima dosta memorije za podatke. Brzo? Memorija je prosto pojedena.

Dobar program mora biti optimalan, a optimalnost niste u stanju da procenite. Na vama je da razmišljate treba li vam još koji indeks, a za meritornu procenu optimalnosti programa morali biste da angažujete komisiju (ugovor o delu) od tri dobra programera koji su ujedno i pošteni ljudi.

Rezime

1. Uvođenjem automatizacije dobijete mnogo, uključujući i nove probleme.
2. Ako već morate da trošite društvene pare na igračke, kupite domine.
3. Ne slušajte tuđe savete, niko ne zna šta sve ne zna.
4. Možete da radite šta hoćete — vaš računar će ipak da se pokvari, a programer će vas sigurno napustiti.
5. Kad primate programera, jedini pravi kriterijum je da li je razveden.

Nonsense in Basic

Ukus čipa

Kakvog je ukusa ULA čip? Da li je bolje zdrobiti ga u male komadiće ili sitno istucati u avanu? Šta od začina treba dodati da bi se lakše svario: biber, beli luk, korijander, origano? Koliko je silicijum opasan po želudac i uopšte po organizam? Postoji li neka stomačna bolest kompatibilna sa silikozom pluća?

Ovakva i slična pitanja Joca mi postavlja svakodnevno. A evo zašto. Opkladio sam se sa njim da ću pojesti „spektrum" na kojem je napisan neki ozbiljan program pomoću „Blasta", „Art Studija" ili nekog sličnog novog uslužnog programa što ima uputstvo na 200 stranica i zauzima 20 K memorije. Možda sam istrčao pred rudu kao neiskusno ždrebe, ali još verujem da sam u pravu. Čak sam čvrsto ubeđen da su svi ti programi velikih mogućnosti praktično neupotrebljivi kad je u pitanju kućni računar od 48 pa i od 128 kilobajta RAM-a.

Nemojte me pogrešno razumeti. Nisam rekao da su to rđavi programi. Ne zameram ništa ni njihovom kvalitetu ni njihovim mogućnostima. Što se toga tiče, sve je OK. Ti programi zaista mogu mnogo i očigledno je da su ih pravili vrhunski programeri. Ali za mene je sve to ipak najobičniji trik softverskih kuća. Moj drug Joca, poznatiji u hakerskim krugovima kao Mister Mikro, glasno tvrdi da sam malo uvrnut, tročoškast i neki levi. Ne veruje mi da su proizvođači softvera morali nečim da pokrenu zasićeno tržište.

Neću reći da sam izuzetak, i ja sam naseo. Međutim, brzo sam se otreznio. „Art studio" sam učitao samo jednom, „Laser Genius" dvaput, koliko da se upoznamo, a od „Blasta" sam odustao posle 46 stranica uputstva. Je programiranja zamišljem kao posao u kome čovek sedi uz računar i kuca po tastaturi (izvinjavam se ljubiteljima miša), a ne u kome sedi uz računar i između svaka dva ukucana znaka pročita pet stranica teksta. O arčenju memorije i da ne govorim. Kakav je to odličan uslužni program kad me zbog nedostatka memorije primorava da svoj program pišem u parčićima umesto u komadu?

Smatram da je po prirodi dobar programer u isto vreme i racionalan duh. Ako nešto ozbiljno programira, upotrebiće uslužni program koji mu daje odrešene ruke, to jest koji ga manje sputava i više čini gospodarom mašine, a ne program koji ga čini robom menija i opcija. Dobar programer uvek može više od bilo kog dobrog programa. Drugim rečima, mnogi dobri uslužni programi su, po meni, namenjeni hakerskoj boraniji, a boranija nikad nije napisala ozbiljan program. Znam to dobro, ne izuzimam sebe iz boranije.

Zato sam se kladio.

Joca još nije uspeo da pronađe ozbiljan program napisan pomoću nekog od pomenutih genijalnih uslužnih. Rok za opkladu već ističe, ima na raspolaganju još samo jedanaest dana. Naravno, niti ću ja stucati čip, niti će Joca bariti u ekspres-loncu kompletno uputstvo za „Laser Genius", mi to samo onako...

Bata Soft

Spectrum plus 3

„Duga" sa tri plusa

Spektrum zlata treba da uđe u škole — uz njegovu bi pomoć učenici začas razumeli teoremu o komutativnosti, asocijativnosti i distributivnosti sabiranja! Plus po plus, Alan Šugar komutira njegove osobine i, distribuirajući razne verzije, zahvata dobar deo evropskog računarskog kolača. Zli će jezici, najzad, reći da „spektrum" sa pridodatom disk jedinicom asocira na pravi kompjuter!

Iako se o njemu govorka već nekoliko meseci „spektrum Plus 3" je ugledao svetlost dana tek prvih dana juna. Alan Šugar, na sreću nije Klajv Sinklar — prvi primerci računara su već u prodaji, što znači da će britanski kompjuterski časopisi uskoro detaljno testirati stari „spektrum" koji je po ko zna koji put obukao novo odelo. U međuvremenu su nam na raspolaganju samo podaci koje je dao sam proizvođač i koji su kao takvi vrlo povoljni.

Kilometarsko ime...

Staro je pravilo da se ime nekog proizvoda koji doživljava mnoge generacije neprijatno produžava: novi Amstradov kompjuter se, punim imenom i prezimenom zove *Sinclair ZX Spectrum Plus 3 128K* (podseća li vas ovo ime na roman „Karavan robova"?). Vlasnik ovako dugog imena se međutim ne razlikuje previše od svog godinu dana starijeg prethodnika — okrenite četvrtu stranu „Računara 20" i ugledaćete nešto sasvim slično ilustracijama koje ovde donosimo. Jedina primetna razlika je naravno i najbitnija: kasetofon je zamenjen diskom!

...i duga istorija

Istorija medija masovne memorije za „spektrum" je kao i istorija samog „spektruma" vrlo interesantna. Spektrum je, sasvim u skladu sa davnom 1982. godinom, u početku koristio kasete. Sinklerov kasetni interfejs je začudo bio sasvim solidan — brzinu od 1500 bauda nije dostigao ni jedan konkurent (setimo se samo „komodora 64"!) dok se na pouzdanosti nije ništa izgubilo: retko ko je imao problema sa programima koje sam snimi. U piratskim sredinama poput naše svaki vlasnik „spektruma" se jasno, bezbroj puta namučio sa kasetom koja je snimljena na nekom drugom kasetofonu, ali su slični problemi prosto-naprosto neizbežni.

Sinkler je međutim, odmah najavio mikrodrajv obećavajući da će njime zameniti skupe disk jedinice baš kao što je „spektrumom" zamenio skupe računare. U početku se o mikrodrajvu mnogo diskutovalo, a on je mnogo kasnio. Kada se konačno pojavio, kritike su bile katastrofalne: radilo se o brzoj kaseti koja se lako kidala uz katastrofalne posledice. Prvi mikrodrajvi su povučeni iz prodaje i nastupilo je novo dugo čekanje. Druga generacija je bila mnogo bolja i spektrumovci je, sa istorijske distance, smatraju osrednjim medijem — pouzdan je i obezbeđuje brzo unošenje programa, ali je pretraživanje kaseta sporo i kao takvo, neprimenljivo za bilo kakve poslove. Mikrodrajv je dalje, previše skup — vlasnici drugih kompjutera za sličnu sumu nabave mnogo upotrebljiviju disk jedinicu! Ponuda softvera prilagođenog mikrodrajvu je, najzad, od početka bila nikakva — softverske firme su zaključile da je interfejs 2 dokupio premali broj spektrumovaca. Igre sa kaseta, jasno, mogu da se prenesu na mikrodrajv ali je ova operacija nedostižna za većinu korisnika: treba se izboriti sa zaštitama, pomeranjem memorije, preklapanjem blokova.

Nezavisne firme već par godina nude razne disk jedinice prilagođene „spektrumu", najveći ugled uživa Opusov *Discovery* koji smo u „Računarima" pominjali već nekoliko puta. Iako je veoma kvalitetan i relativno jeftin, *Discovery* nije mogao da uđe u široku upotrebu: malo vlasnika diska znači malo programa na disketama, malo programa na disketama znači malo vlasnika i krug se zatvara. Neophodno je, dakle, bilo da proizvođač računara odabere neki zvaničan standard što je Amstrad upravo uradio — sastavni deo „spektruma Plus 3" je disk jedinica od 3 inča. Disketama se pristupa samo sa jedne strane koja je podeljena na četrdeset staza; svaka se staza sastoji od 9 sektora po 512 bajta, što znači da na disketu staje tačno 180 kilobajta informacije. Ukoliko vas ne mrzi da ručno prevrćete disketu, na nju ćete upisati dvostruko više podataka — oko 350 K.

Disk jedinice od 3 inča su jedna od veoma retkih očiglednih greški Alana Šugara — poznati biznismen se opredelio za njih, a ostatak kompjuterskog sveta za disk jedinice od 3.5 inča. Posledice su vrlo neprijatne: diskete se teško nalaze i prilično su skupe, podaci snimljeni na nekom Amstradovom modelu se teško prenose na druge mašine (čak i na sam „amstrad PC"), servisiranje je otežano... Dobra strane ovakve odluke je činjenica da su proizvođači disk jedinica od 3 inča voljni da se što pre oslobode zaliha, što znači da je Amstrad u početku bud-zašto dobijao ove osetljive i (često) skupe mehanizme. I ova se prednost, međutim, istopila — sada je sve manje proizvođača „propalih" disk jedinica, što znači da njihova cena neminovno raste. Ovakve su pojave, uzgred budi rečeno, sasvim uobičajene u inženjerskoj praksi: ako vam „crkne" EPROM 2708 (1 kilobajt) u nekom starom računaru zamenu ćete platiti pet puta više nego EPROM 27256 (32 kilobajta)!

Zašto je onda Alan Šugar odlučio da u „spektrum Plus 3" ugradi disk jedinicu od 3 inča? Verovatno da bi održao minimum kompatibilnosti sa ostalim svojim proizvodima — podatke snimljene na „amstradu 6128" možete da učitate na „spektrumu plus 3" i obratno! Za trenutak ćemo videti da je ugradnja ove disk jedinice otvorila i neke CP/M apetite.

Pogled sa strane

Spektrum Plus 3 je opremljen tastaturom koja jedva zaslužuje prelaznu ocenu: izgleda lepo ali, poput svih jeftinih membranskih tastatura, nije za ozbiljnu upotrebu. Čudno je takođe, što na tasterima nisu upisane komande koje se njihovim pritiskanjem dobijaju — ako radite u modu 48 bićete jednostavno slepi! Obrazloženje za ovu besmislicu je, navodno, u tome što Amstrad pretpostavlja da će se svo programiranje na novom računaru obavljati u modu 128 dok će se mod 48 koristiti samo za startovanje igara. Ukoliko se, dakle, upustite u kupovinu novog „spektruma" planirajući saradnju sa prijateljima koji imaju „obične" verzije, pripremite se da naučite napamet ogromnu tabelu ili da pored računara držite nekoliko kolor slika. Ili ćete možda kupiti neku od profesionalnih tastatura?

Vlasnike starog, dobrog i gumenog „spektruma" nisu mnogo zabrinjavali priključci — ne možete pogrešiti priključujući antenski kabl, napajanje i kasetofon! Zadnja strana novog „spektruma" je, međutim, krcata priključcima: tu je standardni Centronics, RS 232/MIDI, RGB, kompozit video, UHF, audio in/out, dva porta za džojstika, port za kasetofon i dodatne disk jedinice. Tu je, pre nego što zaboravimo i *reset* taster.

Čini nam se da je „spektrum plus 3" opremljen svim interfejsima koji će vam



ikada zatrebati. I dalje možete da učitavate softver sa trake, možete da priključujete druge disk jedinice i kopirate programe povezivanje sa štampačem više ne zahteva nikakav interfejs ili skupi kabl nedostaje jedino *on/off* prekidač sa kojim je Sinkler ratovao tokom čitave svoje duge i (povremeno) uspešne karijere. Nije, naravno, predviđeno priključivanje mikrodrajva (šta će vam mikrodrajv kada imate disk?), ali pretpostavljamo da bi softver sa mikrokasete mogao da se prebaci na diskete posredstvom RS 232 kabla ili porta opšte namene.

Modifikovani ROM

Kada uključite novi računar, na ekranu se pojavljuje poruka *(c) 1982 1986 1987 Amstrad PLC, Drives A and M available* i meni koji nudi četiri opcije *Loader, Plus 3 Basic, Calculator* i *48 Basic Loader* je prilično čudan jer omogućava samo učitavanje programa sa diska (treba, jasno, otkucati i ime) dok se stare igre mogu startovati jedino ulaskom u *48 Basic* i kucanjem legendarnog LOAD Plus 3 Basic je sasvim sličan starom 128 K bejziku, s tim što je umesto kasetofona podržan disk, komande su slične AMSDOS-u dok je unutrašnja struktura i dalje nepoznata. O kalkulatoru smo, najzad, već dosta pričali — pogledajte na primer „Računare 15".

Kako stoje stvari sa kompatibilnošću? Pomalo sumnjivo: ulazak u 48 K mod ne garantuje da će sve igre raditi jer su poruke u ROM-u produžene a neke adrese izmenjene. Amstrad se, sa druge strane, trudio da minimizira izmene (nije se, naravno, mogao pomiriti sa ostavljanjem Sinklerovog imena) što znači da će većina igara savršeno raditi, ako neka i krahira, pronaći ćete deset sličnih! Za Jugoslovene je povoljno što je većina kod nas kružećih programa „razvaljena", što znači da su sve provere ROM-a davno eliminisane.

U okviru materijala koji su nam stajali na raspolaganju u toku priprema ovoga napisa, na žalost, nismo pronašli ni jednu reč o važnom problemu interakcije diska i *48 Basic-a*: mogu li se programi sa diskete startovati pri radu sa starim „spektrumom"? Ukoliko mogu, kako računar razlikuje kasetu od diska (možda se za komunikaciju sa diskom koriste komande rezervisane za mikrodrajv?)? Ukoliko ne mogu, „spektrum plus 3" je sasvim bezvredan računar — disk može da se koristi samo za rad sa Plus 3 bejzikom koji je potpuno nepodržan od softverskih firmi. U prospektima se, istina, tvrdi da će svi programi pisani za „spektrum plus 128 K" raditi i na modelu „plus 3" ali je broj takvih programa i dalje zanemarljiv.

„Spektrum plus 3" je, najzad, potencijalna CP/M mašina. CP/M kompatibilnost je za sada ograničena na učitavanje podataka koji su snimljeni na „amstradu CPC 6128" ali je firma *Locomotive Software* navodno postigla dogovor sa *Digital Research-om* o objavljivanju pravog i potpunog CP/M-a „plus 3". O ovome ćemo obećanju, međutim, govoriti tek kada postane realnost — treba saznati da li će na jedvite jade kompletirana biblioteka CP/M programa za CPC 6128 funkcionisati i na novom „spektrumu".

Skupo i sumnjivo

„Spektrum plus 3" tranutno košta 250 funti, što je prava pljačka — pomislite da se stotinak funti više možete da nabavite „*amstrad PC*" sa diskom monitorom i 512 kilobajta RAM-a! Verujemo da bi tek se stotinak funti niže cena mogla da učini „Plus 3" konkurentnim.

Čak i kada bi cena bila konkurentna, postavlja se pitanje upotrebljivosti novog računara. Šugar, manirom iskusnih poslovnih ljudi, već par godina varira karakteristike jednog „antičkog" računara, ali pri tome pažljivo izbegava da učini „spektrum" poslovnom mašinom kako su stvari počele, u „Računarima 100" (koji, ako je verovati Dajanovim pitalicama, izlaze negde oko 2000. godine) ćemo prikazati *Sinclair ZX Spectrum Double Plus 17 4 M* koji će biti opremljen sa dva hard diska od po 135 megabajta, ali koji će i dalje u svakom redu ekrana ispisivati po 32 slova! Ako vam je dakle, potreban računar za igre, i dalje bismo vam savetovali običan „spektrum". Ali, Šuger se i toga dosetio — običan „spektrum" se više ne pravi.

Dejan Ristanović

„Spectrum plus 3" tehničke karakteristike

1	Microprocesor	Z80
2	Clock	3 5 MHz
3	ROM	48 K
4	Sadržaj ROM-a	48 Bejzik Plus bejzik
5	RAM	128 Basic Calculator 128 K
6	Tastatura	membranska 58 tastera
7	Ekran	TV monitor
8	Tekst	32 * 24
9	Grafika	256 * 192
10	Spoljna memorija	flopi disk 3 inča SS/DD 40 traka
11	Kapacitet diskete	180 K po strani
12	RS 232	da
13	Centroniks	da
14	Časovnik realnog vremena	ne
15	Džojstik portova	2
16	Ostali interfejsi	audio in/out Midi kasetofon drugi disk, port opšte namene
17	Cena	250 funti (računar, džojstik 6 igara)

Štampači sa dvadeset četiri iglice

Epson to radi bolje

Već je gotovo postala tradicija da u letnjim brojevima „Računara" prikazujemo nove matrične štampače. Ova je tradicija, pre svega, zasnovana na činjenici da su godišnji odmori vreme kada mnogi naši zemljaci putuju u inostranstvo i razmišljaju o kupovini raznoraznih računara i periferijske opreme. Pomažu nam, uz to, i proizvođači štampača koji približno jednom godišnje liferuju čitavu novu generaciju — pre tri godine su to bili Epson RX i FX 80, Seikosha GP100A i Star Gemini 10 X, pre godinu dana Epson LX80 i LX85, Canon PW 1080A, Mannesmann Tally 85, Panasonic KX-P1092 i Star SD10, a ove godine 24-pinski Epson LQ 2500, NEC P6, Toshiba P321 i Star NB-15.

Kvalitet otiska je i dalje osnovni parametar po kome se procenjuje vrednost nekog štampača na kraju krajeva. čak i ako imate super personalac sa 32-bitnim mikroprocesorom, dva hard diska po 130 megabajta i Sinklerov štampač, svaki će vaš korespondent i saradnik zaključiti da ste se opremili „spektrumom"! Najbolji otisak nude znamo štampači sa lepezom i pisaće mašina sa kuglicom. Ovakvi printeri, na žalost, nisu osposobljeni za definisanje znakova, dampovanje ekrana i mnoge slične operacije koje su veoma potrebne u svakodnevnom radu. Kupovina dva štampača je i dalje prilično retka pojava — ne samo da je cena visoka već se postavlja pitanje povezivanja opreme softverskih drajvera i, naravno, prostora na uvek pretrpanim stolovima. Kvalitetan matrični štampač je, zato, solucija za koju su se mnogi opredelili, pogotovu od vremena kada su jeftini matrični štampači dobili NLQ (*Near Letter Quality*) mod. NLQ slova su daleko lepša od standardnih ali, istini za volju, nisu uporedive sa proizvodom *daisy wheel* štampača. Osim toga NLQ pisanje je strahovito sporo, tako da se korisnici u većini situacija opredeljuju za standardna slova.

Skok bez prestupa

Svi dobro znamo da se rad matričnih štampača zasniva na koloni iglica koja udaraju u specijalnu traku. Obzirom da slovo nastaje kolonu po kolonu i da se sastoji od tačkica, čitljivost može da bude neprijatno slaba, pogotovu za ljude koji nisu navikli na ovakav način pisanja. Struktura će se jasno manje primećivati ako su tačke gušće raspoređene, što znači da štampači koji imaju više iglica proizvode kvalitetniji tekst. Do skora su glave štampača imale devet iglica pa je slovo nastajalo na matrici 11*9. Minijaturizacija komponenata je, međutim, omogućila precizniju konstrukciju glava i tako su nastali 24-pinski štampači.

Ukoliko vam se skok sa 9 na 24 učinio prevelikim, unakoliko ste u pravu: glava 24-pinskog štampača zaista ima 24 iglice ali su one raspoređene u dve veoma bliske kolone od po 12 iglica. Desna kolona je malo smaknuta u odnosu na levu, što znači da su iglice raspoređene u cik-cak. Rastojanje između pina 1 i pina 24 je jednako rastojanju između prvog i devetog pina na starim modelima, što znači da je veličina slova jednaka, bitno su povećana rezolucija, kvalitet otiska i, na žalost, cena.

Pominjanje viših cena ne mora da znači da su 24-pinski štampači nedostižni za prosečnog korisnika — cena ovih uređaja je sasvim približna nekadašnjim 9-pinskim Epsonovim modelima — 500—600 funti za 80-kolonske i 1000 funti za 132-kolonske printere. Iz tabela se vidi da NEC P6 i „tošiba P321" pripadaju prvoj, a „epson LQ 2500" i „star NB-15" drugoj grupi. Sva ova četiri štampača su praktično jednaka u suštinskim detaljima, čemu se ne treba preterano čuditi: svi su modeli „epson" kompatibilni, što znači da programi koji definišu znakove na „epsonu" moraju da rade i na njima.

Kontrolni panel

Svaki se štampač, jasno, nalazi pod kontrolom računara: umesto slova, kompjuter može da pošalje propisanu sekvencu kontrolnih kodova i tako izazove promenu tipova slova, prelazak na novi list i slične stvari. Kontrolisati štampač posredstvom računara nije, međutim, baš uvek komforno: često ćemo poželeti da preskočimo nekoliko redova ili pređemo na sledeći list a da pritom ne pamtimo komplikovane 'Escape' sekvence i ne kucamo duge komandne linije. Zbog toga je na kutije svih modernih štampača ugrađeno po nekoliko tastera. Uz neizbežni *On-Off*, štampači koji slede Epsonova standarda imaju dirku *On Line* pomoću koje se štampaču zabranjuje da prima znakove od računara (zbog čega je to korisno? Ponekad će vam zatrebati da promenite list papira ili obavite telefonski razgovor u toku koga vas buka neće ometati), taster *Line Feed* pomoću koga štampač preskače jednu liniju i, najzad, taster *Form Feed* koji izaziva prelazak na prvi red sledeće stranice. „Epson LQ 2500" je, uz to, opremljen tasterom koji obezbeđuje privremeni prelazak sa običnog na LQ ispisivanje, ponekad neke strane teksta treba da izgledaju posebno lepo, dok neke druge nisu posebno bitne!

Tasteri poput *On line*, *Line Feed* i *Form Feed* su možda bili dovoljni pre nekoliko godina, ali su apetiti korisnika u međuvremenu bitno porasli. Tako je Epson još kod svoje LX serije uveo takozvani *SelecType* — pritiskanjem odgovarajućih kombinacija tastera kontrolnog panela korisnik može da izabere raznorazne tipova i širine slova i uopšte da sebe poštedi bilo kakvih *escape* sekvenci. Nevolja sa *SelectType*-om je što sekvence nije lako popamtiti i što čovek nikada ne zna šta je uradio — štampač se, istini za volju, oglašava raznim tonovima (sećate li se Artu-Ditua?) ali njihove kombinacije nije uvek lako razumeti.

„Epson LQ 2500" ima panel od četiri tastera i alfanumerički LCD ekran od 20 znakova na kome u svakom trenutku možete da pročitate trenutno stanje štampača. Ovo stanje, pritiskajući tastere, po želji menjate pri čemu se svaka promena odmah reflektuje na ekranu — zvučnik se oglašava samo kada pogrešite! Ovaj takozvani *SelecType 2* će neobično obradovati kako početnika tako i iskusne korisnike — prvi će morati manje da znaju, a drugi će do željenih rezultata dolaziti mnogo brže.

Sledeći korak napred koji je LQ 2500 napravio je potpuno odsustvo mikroprekidača. Bezmalo svi dosadašnji štampači su, uz kontrole na kućištu, imali i grupe mikroprekidača uz pomoć kojih se postavlja početno stanje printera. Ukoliko, na primer, uglavnom radite sa kondenzovanim slovima, jednom ćete odvrnuti nekoliko šrafova, skinuti kućište štampača i pomeriti mikroprekidač koji se odnosi na kondenzovani mod. Tako ćete izbeći slanje kontrolne sekvence koja aktivira kondenzovana slova po svakom uključivanju štampača ali ćete, kada vam jednom zatrebaju normalna slova, morati da šaljete drugu sličnu sekvencu ili da čeprkate po panelu. Opšta karakteristika mikroprekidača je, dakle, da se stanje koje je njime postavljeno *može* poništiti ili promeniti softverski.

„Epson LQ 2500" je, kako rekosmo, ukinuo mikroprekidače i zamenio ih EEPROM-om. EEPROM je skraćenica od *Electrically Erasable Programmable Read Only Memory* ili, u prevodu, ROM koji se elektronski briše. Kada ste, dakle, zadovoljni stanjem koje ste postavili na panelu, pritisnete taster i ovo se stanje upisuje u EEPROM, štampač će po svakom sledećem uključivanju biti konfigurisan na isti način. Još su zanimljivije četiri takozvane makro definicije — ukoliko štampač koristi više ljudi, svako može da zada početno stanje koje odgovara njegovim potrebama i da ga docnije poziva pritiskom na svega par tastera. Početno stanje i svi makroi mogu po želji da se ispišu na LCD ili, što je posebno zanimljivo, na papiru.

Šta se sve podešava pomoću panela, odnosno mikroprekidača? Za nas je najvažnija mogućnost izbora dužine stranice (naš list ima 72 reda, a engleski i američki 66), početnog seta karaktera (mi obično prerađujemo švedska u jugoslovenska) i kontrola





***Savršenstvo jedne tehnologije** Printeri sa 24 iglice u glavi za štampanje donose novi standard u kvalitetu otiska i predstavljaju vrhunac klasične tehnologije u proizvodnji štampača*

1 MODEL	Epson LQ-2500	NEC P6	Star NB-15	Toshiba P321
2 Orijentaciona cena (funti bez VAT-a)	995	549.	949	590
BRZINA				
3 Obična slova, 10 cpi	270	180	250	180
4 Obična slova, 12 cpi	324	—	300	216
5 NLQ slova, 10 cpi	90	60	83	80
6. NLQ slova, 12 cpi	108	—	100	72
KONTROLNI PANEL				
7 Izbor NLQ mode	da	da	ne	ne
8 Izbor tipa slova	da	ne	ne	ne
9 Taster 'Top of Form'	ne	ne	ne	da
PAPIR				
10. Perforirani papir	da	opcija	da	opcija
11 A4 listovi	da	da	da	da
12 Rolne	da	da	da	da
13 Min širine (mm)	102	114	102	102
14 Max. širine (mm)	410	254	384	280
15 Karbon kopija	3	3	3	3
UPOTREBA				
16. Bafer (KB)	8	8	16	2
17 Različitih fontova	7	3	3	3
18 Slova u redu (najviše)	272	170	244	132
KOMPATIBILNOST				
19 Serijski interfejs	da	opcija	opcija	opcija
20. IBM kompatibilnost	da	da	da	da
ODRŽAVANJE				
21 Cena trake (funti)	8 10	8 30	8	11
22 Proklamovani vek trake (miliona znakova)	2	3	4 5	2
UPUTSTVO ZA UPOTREBU				
23 Tomova	2	2	2	1
24 Kvalitet	odličan	dobar	dobar	OK

preskakanja perforacije na kraju svake stranice. Ni jedan od štampača koji pominjemo nema mogućnost da detektuje perforaciju između stranica, što je odlika nekih skupih profesionalnih modela: kraj strane se prepoznaje tako što je štampaču poznata njena dužina i time što pretpostavlja da se po svakom uključivanju nalazi na početku novog tabaka. Ukoliko vam se, dakle, dogodi da isključite štampač koji nije završio sa ispisivanjem stranice, moraćete da okrećete papir pomoću uvek prisutne ručice i da uključite printer tek kada perforacija prođe ispod glave. Tako smo u prilici da pomenemo jedinu ozbiljnu manu „epsonovog" panela — nedostaje taster *Top of Form* koji bi ukinuo potrebu da gasimo i palimo printer kada, posle neke greške, postavimo papir na početak nove stranice. U nedostatke bi mogla da se ubroji i nemogućnost (ili bar nedokumentovanost) softverske promene sadržaja EEPROM-a (probajte ESC „*"). Skloni smo da verujemo da će neki od sledećih modela biti osposobljen za dvosmerno opštenje sa računarom koji bi mogao da pročita i promeni početno stanje, premda bi za nešto slično bio neophodan dvosmerni Centronics interfejs.

„NEC P6", „tošiba P321" i „star NB-15" nemaju ovoliko moćne panele, pa su opremljeni mikroprekidačima koji su, na sreću, lako pristupačni — treba samo malo pomeriti glavu. „Star NB-15" omogućava izbor tipa slova sa kontrolnog panela, dok su komande na preostala dva modela ograničene na standardno *Line Feed* i *Form Feed*. „tošiba" ima i taster *Top of Form* koji, kako rekosmo, „epsonu" nedostaje.

```
This is a sample of draft mode printing
THIS IS A SAMPLE OF DRAFT MODE PRINTING.
(The box characters are all in NLQ mode)
This text is now in NLQ,which is slower.
THIS TEXT IS NOW IN NLQ,WHICH IS SLOWER.
(All the print samples are actual size)
```

„Epson LQ 2500"

```
This is a sample of draft mode printing
THIS IS A SAMPLE OF DRAFT MODE PRINTING.
(The box characters are all in NLQ mode)
This text is now in NLQ,which is slower.
THIS TEXT IS NOW IN NLQ,WHICH IS SLOWER.
(All the print samples are actual size)
```

„NEC P6"

```
This is a sample of draft mode printing
THIS IS A SAMPLE OF DRAFT MODE PRINTING.
(The box characters are all in NLQ mode)
This text is now in NLQ,which is slower.
THIS TEXT IS NOW IN NLQ,WHICH IS SLOWER.
(All the print samples are actual size)
```

„Star NB-15"

```
This is a sample of draft mode printing.
THIS IS A SAMPLE OF DRAFT MODE PRINTING.
(The box characters are all in NLQ mode)
This text is now in NLQ,which is slower
THIS TEXT IS NOW IN NLQ WHICH IS SLOWER
(All the print samples are actual size)
```

„Tošiba P321"

Brzina i bafer

Brzina je jedan od veoma važnih parametara pri izboru štampača, što početnici retko priznaju. autor ovoga teksta je, na primer, svojevremeno kupio „seikošu GP 100A", smatrajući da mu njena sporost i buka neće previše smetati — ocena je bila odgovarajuća! Već je prvi eksperiment pokazao koliko je čitava logika bila pogrešna — štampač je pravio dreku kao parni cirkular i ta je dreka trajala satima čak i kada se štampao tekst prosečne dužine. Rezultat — sa pola godine smo prešli na „epson"! Učite se, đaci, na tuđim greškama i pažljivo proverite brzinu modela koji nabavljate. Nevolja je što niko na svetu ne zna kako proizvođači štampača dolaze do brzine koju navode u specifikacijama. realne su brzine bitno manje (priča se da se teorijske brzine odnose na kondenzovani mod i da je svako vreme prepolovljeno navodno da bi se obračunalo pomeranje papira). Zato smo na slici i prikazali rezultate sintetičkog brzinskog testa — štampan je sadržaj datoteke sa programom koja je duga 73500 bajta i to u najsporijem i najbržem modu.

Iako se još nismo detaljno bavili LQ odnosno NLQ modom, podaci iz tabele će vas prilično začuditi — Epsonove LX serije, „kanon PW 1080", „panasonik 1090" i slični štampači su u svakoj sekundi ispisivali stotinak običnih ili desetak NLQ znakova, dok su kod 24-pinskih printera razlike daleko manje — „epson LQ 2500" u NLQ modu postiže brzinu starih modela iz serije RX koji su ispisivali samo obična slova. Ovaj dobitak u brzini je neposredna posledica nove tehnologije — 24-pinska glava ne mora dva puta da prolazi preko istog teksta, što znači da se povećava jedino horizontalna rezolucija.

Početnike će verovatno začuditi što se vreme koje smo nazvali *slobodan računar* razlikuje od vremena *slobodan štampač* — zar računar nije zauzet u toku čitavog štampanja? Verujući da smo u toku poslednje godine dobili dosta novih čitalaca, posvetićemo nekoliko redova onome što se događa od momenta kada računar pošalje jedan znak štampaču do momenta kada se taj znak pojavi na papiru. Veze računara i štampača se obično zasnivaju na principu takozvanog *hand shaking*-a: osim linijama za prenos podataka, računar i printer su povezani linijom koju zovemo *ready-busy*. Preko ove linije štampač signalizira računaru da je spreman da primi sledeći znak koji mu zatim biva i poslat. Znak putuje pravo u takozvani *bafer*, specijalni RAM koji je ugrađen u štampač. Kada se bafer popuni (njegova veličina varira od modela do modela), računar mora da čeka da se neki znak odštampa da bi u baferu bilo mesta za sledeći. Veliki bafer čini da se računar „oslobodi" za druge poslove mnogo pre nego što se štampanje završi, ukoliko uglavnom pišete kraće pisma i izveštaje, isplatiće vam se da investirate u veći bafer jer će tako štampač moći da radi dok vi (ako vam ne smeta buka) pišete sledeće pismo. Na baferu se, sa druge strane, može dosta i uštedeti: računari sa pristojnim RAM-om koji su opremljeni fleksibilnim operativnim sistemom mogu da organizuju komunikaciju sa štampačem preko prekida (interapta) i tako paralelno obavljaju dva posla koristeći višak sopstvenog RAM-a kao bafer za štampač. Primer ovakve metodologije je program PRINT koji vlasnici MS DOS kompatibilnih mašina relativno retko koriste.

Pre nego što se oduševite veličinom bafera nekog od modela koji opisujemo (baferi su ranije imali par kilobajta, a danas dostižu i 16 K pa se čak mogu i dalje proširivati dodavanjem statičkih RAM čipova), razmislite o jednom ograničenju: uobičajeno je da se bafer za štampač koristi i za definicije karaktera tako da se pomeranjem specijalnog mikroprekidača bira jedna od ove dve namene. Kako će vam s vremena na vreme zatrebati naša latinična slova i kako će vas mrzeti da par puta nedeljno razmišljate o mikroprekidačima velikog bafera ćete se najverovatnije zauvek odreći!

Fontovi, modovi i širine

Pošto je neki karakter iz bafera došao na red za ispisivanje, printer će ga analizirati i konstatovati da li se radi o običnom znaku ili o kontrolnom kodu. Ukoliko se radi o znaku, treba još razmisliti o načinu na koji će se on što brže preneti na papir. Svi štampači koje opisujemo imaju mogućnost štampanja u dva pravca uz takozvano logičko traženje: glava će se, pre nego što počne da ispisuje red, pomeriti na onaj njegov kraj koji joj je bliži.

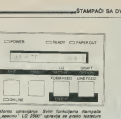
Pronalazak kontrolnog znaka signalizira štampaču da iz bafera uzme i sledeće karaktere koji će zaokružiti takozvanu *Escape sekvencu*. Escape sekvenca je niz kontrolnih kodova koji počinju sa 'Escape' (27) i koji kontrolišu štampač ili mu naređuju da buduće znakove ispisuje na neki specijalan način.

Domaći ljubitelji računara (pa čak i kompjuterski profesionalci) često mešaju termine „tip slova" i „stil" (u stranoj literaturi *font* odnosno *typestyle*). *Tip* označava oblik slova, tj. definiciju znaka koja mora da bude smeštena u ROM štampača. Epson LQ 2500, na primer, ima sedam različitih oblika: *draft*, *italic*, *sans serif*, *courier*, *prestige*, *script* i *letter quality*, posebno je interesantan *script* koji imitira pisana slova. Preostala tri štampača imaju samo po tri standardna oblika: *draft*, *italic* i *letter quality*.

Što se stilova tiče, svi modeli o kojima govorimo mogu da ispisuju kondenzovana (17 znakova po inču), povećana (5 znakova po inču) i kondenzovana povećana slova (8,5 znakova po inču). Stilovi i fontovi mogu slobodno da se mešaju, što znači da tekst možemo da pišemo kondenzovanim kurzivom *(italic)*, a naslove proširenim NLQ znacima. Ne treba zaboraviti proporcionalna, istaknuta, podvučena i dvostruko ispisana slova, verujemo da je značenje ovih termina samo po sebi jasno. Pomenimo, najzad, indekse i izložioce koji su neophodni svima koji *se* bave matematikom i prirodnim naukama.

Vrste papira

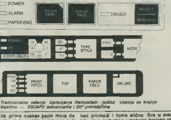
Laicima je najnormalnije da štampači koriste običan papir A4 formata. Ovaj se papir, kao i na pisaćoj mašini, kreće oko pokretnog valjka koji je uključen u standardnu opremu svih modela o kojima govorimo. Nije, međutim, baš prijatno koristiti A4 listove: štampaču je, da bi ispisao jedan takav list, potrebno manje od minuta, što znači da ćete svakoga minuta morati da ubacujete stranice! Posle izvesnog vremena utvrdićete da ubacivanje papira odnosi više vremena nego samo štampanje, pa ćete potražiti racionalnija rešenja. Jugoslovenski vlasnici računara obično koriste perforirani kompjuterski papir (u stranoj literaturi *fan-fold paper)* koji se obično isporučuje u sanducima od kojih svaki sadrži po 2000 međusobno povezanih listova prošivenih perforacijom sa svake strane (sanduk od 2000 listova se „zove" 1+0 24*12 2000 RN 2400 *Bjanko*, košta oko 15000 dinara i autoru ovoga teksta koji mnogo ne štedi papir traje dobrih desetak meseci). Štampač





Komforno upravljanje Svim funkcijama štampača na „epsonu' LQ 2500" upravlja se preko testature na čeonoj strani, a status se može očitati na mini-displeju

Bez velikih poboljšanja Štampač NEC P6, pored osnovnih funkcija, omogućuje se kontrolnog panela i izbor tipa slova



Tradicionalno rešenje Upravljanje štampačem „tošiba" obavlja se krajnje klasično — ESCAPE sekvencama i DIP prekidačima

koji treba da prima ovakav papir mora da bude opremljen takozvanim *traktorom* koji je uključen u cenu „epsona" i „stara", oni koji se odluče za „tošibu" ili NEC će morati da žrtvuju još dvadesetak funti za ovaj neophodni dodatak. Naša tabela prikazuje maksimalnu širinu papira koju pojedini modeli mogu da prima, samo se po sebi razume da 132-kolonski štampači mogu da koriste kako normalan A4 papir tako i uske nizove nalepnica koje su značajne za razne poslovne primene

Najjeftinija rešenja neobično popularno među hakerima jesu role papira koje primaju svi modeli o kojima govorimo Ovaj je papir prilično nekvalitetan i neprimeran mogućnostima 24-pinskog štampača

Ostalo je još da pomenemo uređaja za automatsko umetanje papira koji su namenjeni ljudima koji pišu mnogo poslovnih pisama — kada već imate 24-pinski štampač, red je da pišete na bank postu! Svaki od modela o kojima govorimo može da se dopuni uređajem koji, uz manje ili više problema (manje probleme na „epsonu" mnogo na NEC-u), sam umeće papir i kontroliše „skladištenje" ispisanog teksta Ove su naprave prilično hirovite — listovi se ponekad lepe jedan za drugi, mašine ponekad promaši i tome slično Sve u svemu, iako je automatsko umetanje korisna stvar, ono će morati da se obavlja pod vašom neposrednom kontrolom!

Cepanje papira je veoma bolna tačke za većinu štampača u normalnom je radu često potrebno otcepiti upravo ispisani list papira bez potrebe da se izbacuje još jedan prazan i, po mogućnosti, bez potrebe da korisnik ustaje sa stolice Pitanje cepanja papira je razrešeno na najrazličitije načine i, ma kako banalno izgledalo, predstavlja jedan od veoma ozbiljnih parametara koji treba uzeti u obzir pri izboru štampača Što se modela koje opisujemo tiče, najbolje su se pokazali „tošibe" i, unekoliko „epson"

Svi modeli koje opisujemo mogu da proizvedu određen broj karbon kopija dokumenata Te se kopije, međutim ne prave pomoću „običnog" indigo — treba kupiti poseban papir kod koga se svaka stranica sastoji od više listova između kojih je specijalan indigo (sećate se onoga 1+0 u oznaci papira koji koristimo?)

Interfejsi i kompatibilnost

Svi štampači koje prikazujemo imaju ugrađen paralelni (Centronics) interfejs što znači da ih je lako povezati sa većinom danas popularnih personalnih računara „Epson LQ 2500" je opremljen i serijskim (RS 232) interfejsom, dok je za serijsku komunikaciju sa ostalim modelima neophodan dodatni interfejs koji se ugrađuje u sam štampač, podnožje i konektori su predviđeni a interfejs obuhvata i dopunski bafer Pitanje je, međutim koliko su serijski interfejsi potrebni raširano verovanje da serijski prenos podrazumeva sporije štampanje je potpuno pogrešno jer „usko grlo" u paru računar — štampač nije komunikacija već brzine ispisivanja znakova na papiru Prodavci kompjuterske opreme su uobičajili da uz skupije štampače besplatno daju i kabl za povezivanje sa vašim računarom ali se na ovu konvenciju ne možete baš uvek osloniti za kabl ćete ponekad morati da doplatite 10—20 funti

Svi su štampači koje prikazujemo kad što rekosmo na početku napisa „epson" kompatibilni ali i prilagođeni IBM-ovim standardima Šta je zapravo IBM kompatibilan štampač? ASCII standard se odnosi samo na znakove čiji su kodovi između 32 i 127 dok su tvorci PC-jevog generatora karaktera iskoristili i kodove 128—255 — tu se nalaze strana slova, obična i udvostručena linija kao i razna druga luda slova i znakovi kao što su oznake na „kartama" Epsonov standard sa druge strane koristi kodove 128—255 za kurzivne (*italic*) znake što znači da tvorci IBM/„epson" kompatibilnih printera moraju da predvide kontrolne sekvence koje objašnjavaju štampaču sa kojim je računarom povezan Tošibini inženjeri nisu doveli ovaj posao do kraja — „zaboravili" su udvostručene linije i zamenili ih običnima!

Što se grafike tiče, svi modeli mogu da emuliraju stare „epsone", što znači da će svi programi za dampovanje ekrane raditi savršeno Nama, međutim, mnogo smisla kupovati nov štampač koji će prosto emulirati stari — proverite da li vaša verzija „AutoCAD-a" ili drugih sličnih programa koje koristite podržava 24-pinska ispise ekrana Imajte, takođe, u vidu da veća rezolucija zahteva prenos veće količine podataka, što znači da preplsivanje ekrane na papir može i da potraje, zato će kvalitet biti izvanredan!

Kakva je situacija sa YU znacima? Svi modeli o kojima govorimo imaju RAM sasvim dovoljan za naših 8 ili 10 slova ali će se ambiciozniji korisnici svakako potruditi da znakove ugrade u ROM Od opreme će vam trebati programator koji prima EPROM-e 27256 i 27512, dok će dizajnerski rad biti daleko komplikovaniji — što je rezolucija veća to je teže iscrtati slovo tako da lepo izgleda „Računari" će se potruditi da u nekom od sledećih brojeva opišu ugradnju YU znakova u neke popularne 24-pinske modele

Buka i vibracije

Štampači proizvođeni pre pet-šest godina imaju jednu užasnu osobinu prave nesnosnu buku Obzirom da većina hakera ima običaj da radi noću, nabavka štampača koji neće nikoga buditi i nije tako loša investicija

Konstruktori modernih 24-pinskih štampača su uložili dosta truda da buku svedu na minimalnu meru, tako da su svi modeli koje prikazujemo prilično tihi, čak toliko tihi da *silent mode* nije ni implementiran. Tabelu nismo dopunili našom „Sl" jedinicom za merenje buke zvanom *broj zatvore-*

nih vrata jer se nijedan od modela ne čuje čak ni u susednoj sobi

Usleni štampači donose nov problem — frekvencija zvuka je strahovito visoka što je neprijatno pa čak i nazdravo. Ova je mana posebno prisutna kod „tošibe" koja se čuje čak i kada ništa ne štampa — moraćete stalno da uključujete i isključujete štampač! Sa bukom štampača se meša i buka ventilatora koji hladi glavu — neki modeli imaju čak i po dva ventilatora!

Buka je sve u svemu, najmanja neprijatna kod „stara" i NEC-a" dok se „epson" i „tošiba" nisu baš proslavili. Što se vibracija tiče „star" i „epson" su se najgore pokazali možda zbog toga što su najbrži

Upotreba i uputstvo

Za razliku od računara, štampači zahtevaju određeno održavanje: treba menjati papir, montirati traku, pomerati mikroprekidače, odstranjivati „konfete" koje preostaju u samom štampaču i raditi mnogo sličnih stvari. Za razliku od starijih modela, štampači koje opisujemo su pristupačni za ove svakodnevne operacije, posebno je olakšana promena trake koja vam neće zaprljati ruke. Kada pomenusmo traku, naša tabela prikazuje njenu cenu i proklamovani vek trajanja — trake su vrlo kvalitetne, vrlo skupe i nabavljive jedino u inostranstvu

Glave 24-pinskog štampača je prilično skupa, što znači da ćete, ako se vaše iglice izlížu, morati da potrošite nekih 150 funti. Proklamovani vek glave je, međutim, vrlo dug, što znači da će ovom trošku biti izloženi samo najneaaraćniji među čitaocima ovoga teksta

Vremena kada se uz štampače dobijalo katastrofalno slabo uputstvo po svemu sudeći pripadaju ružnoj prošlosti — uz svaki od ovih modela (osim „stara") dobijate dve knjige od kojih jedna predstavlja uvod u svet štampača a druga tehničko uputstvo za upotrebu. Posebno su kvalitetne knjige koje daje Epson, dok je Toshibino uputstvo vizuelno vrlo atraktivno ali ne i preterano korisno. Mnogi korisnici računara koji se ne bave programiranjem, međutim, i dalje imaju mnogo razloga da se žale na dokumentaciju — svi su primeri pisani u Microsoftovom bejziku, dok podataka o drajverima za WordStar, WordPerfect i slična programa nema ni za lek. Vredi napomenuti da je naš umetak „Štampači" iz „Računara 16" sasvim primenljiv za sve modele o kojima govorimo pri čemu bi ga trebalo dopuniti podacima o poboljšanoj grafici i definicijama NLQ znakova

Sve u svemu, verujemo da su 24-pinski štampači povratili „epsonu" staru slavu — LQ 2500 je očito superioran svim drugim modelima iz iste klase. Osnovu za ovaj zaključak predstavlja izvanredan panel sa ekranom, veliki broj ugrađenih fontova i kvalitet otiska koji smo, u nemogućnosti da nabavimo sve modele, preneli iz časopisa *Personal Computer World*. Dobru alternativu predstavlja „star" koji je izuzetno brz i jeftin, dok je NEC P6 očito superioran „tošibi" koju bismo vam teško preporučili

Nova generacija 24-pinskih štampača verovatno predstavlja vrhunac jedne uspešne tehnologije — ne verujemo da će se u budućnosti broj iglica i kvalitet otiska bitno poboljšavati. Korisnicima kojima 24-pinski štampač nije dovoljan ostaje jedino da razmišljaju o laserskim printerima.

Dejan Ristanović



Peek & poke show

Hakersko proleće u Londonu

Svi oni koji su se potajno nadali (to se ne sme glasno) da će Peek & poke show daleko dogurati nisu bili daleko od istine. Peek & poke show je dogurao do Velike Britanije. Malo se prošetao, pokupio utiske i slatki ukus računarske sredine i vratio se da referiše. Svojim vernim čitaocima, naravno. Zbog preokupacije svetskim računarskim pitanjima, domaća u ovom broju ostavljamo po strani. Od sledećeg broja sve po starom.

Proleće je već na izmaku i to je već samo po sebi dovoljan razlog da se čovek prošeta malo po (računarskoj) Velikoj Britaniji i pogleda kako stoje stvari u domovini kućnog računarstva. Da ovo ne bi bio još jedan prikaz zagledanja u izloge i uzdisanja nad novitetima (kojih trenutno i nama mnogo) neka ovaj izveštaj bude u obliku šetnje se sedam teza ili teza u sedam koraka

Prvi korak ili teza o iščekivanju

Kao i bilo ko drugi inficiran računarima na ovaj ili onaj način, vaš verni šetač je hteo da se odmah upozna sa tim šta se događa. Sa računarima u Britaniji, naravno. Najbliža prodavnica novina nudila je petnaestak računarskih časopisa. Zahvaljujući ljubaznom prodavcu (valjda se na meni vidi da sam Jugosloven) i obilnim količinama moje drskosti prelistao sam u dva i po sata sve te časopise i pročitao nekoliko najzanimljivijih. Toliko informacija! Toliko sitnih lepih stvari koje se događaju. Onda, kad se malo bolje razmisli, i ne događa se baš ništa naročito. Nema velikih vesti. Ako se pažljivo zagledaju uvodnici i komentari, može se nekako skrpiti teza o tome da svi iščekuju da se ove godine nešto značajno i desi. Opšta PCacija i onako nije neka novost. Zato se može reći da je glavna novost nedostatak novosti i neuobičajeno zatišje

Drugi korak ili kako se zaposliti

Kada Englez hoće da se zaposli on ode u svoj lokalni Job Center i pogleda kakvi se poslovi nude (to baš nije tehnika za najjeftinije i najskuplje poslove). Detaljni spiskovi ponude i potražnje na tržištu radne snage mogu se dobiti unutra, ali je ipak lako videti šta je hit. Najzanimljivija ponuda se stavlja ju direktno u izlog na neku vrstu panoa. Čega ima najviše? Računara! Hit sezone su upravo ljudi koji upotrebljavaju računar u nekom standardnom poslu. Istina, još uvek se dosta traže programeri i sistem analitičari kao i razne klase operatera, ali su primat (i skoro čitave izloge) dobila trivijalna zanimanja — od sekretarice za kompjutersku obradu podataka do generalnog direktora koji obavezno mora da zna kako se koristi Managerial System. Plate? Plate su taaaako zanimljive da ih ovde ne treba pominjati. Zašto izazivati? (Činjenice za razmišljanje: centri za dobijanje posla preporučuju izvesnu vrstu kompjuterskog obrazovanja ili doobrazovanja svima koji nameravaju da traže posao preko njih od zidara do hemičara. Aha?)

Treći korak ili mase i klase i tako to

Pravi računardžija ne sme da se iznenadi kada vidi da neko koristi računar ma kako neobična okolnosti bila. Tako je bilo sasvim prirodno kada je čovek koji je sedeo do mene u avionu izvadio iz svoje akten tašne mali Hjulit-Pakardov prenosivi računar i gomilu papira i počeo da radi bez odlaganja. To su stvari koje su uobičajene. Bar u poslovnom i poslovnijem svetu. Ono što je zaista bilo iznenađenje je bilo otkriće da sve prodavačice u velikoj „Marks & Spencerovoj" robnoj kući imaju po jedan „Psion Organiser". Svaku od njih! išao sam od jedne do druge i kontrolisao. Koriste ga za proveru cena, za kontrolu zaliha i za gomilu drugih sitnih poslova. Obišao sam još nekoliko „Marks&Spencerovih" robnih kuća u Londonu i još par komada u Redingu, Oksfordu i Vindzoru. Sve njihove prodavačice imaju „Psion Organiser". Svaka čast domaćina!



Četvrti korak ili samo baciti pogled

Šta kaže površno bačen pogled ako se čovek koncentriše na računare. Kao prvo — računari su definitivno prodrli u medije. Skoro svi filmovi koji su trenutno hit u Londonu (ako ne računamo Tarkovskog) imaju na špici malu oznaku "Special efects by Industrial Light and Magic". Dakle kompjuteri! Skoro sve reklame na televiziji su ili kompjuterski generisane i obrađene ili čak i sadrže kompjuter kao deo scenografije. Sve novine i časopisi su puni oglasa za računare i računarsku opremu. To ne mimoilazi ni dečje stripove, ni dostojanstveni Financial Times. Što se škole tiče upravo je u toku velike rasprave o tome da li je ono što je konzervativna vlada učinila da uvede dosledno kompjutere u obrazovanje pa tima i u život, dovoljno. Opšti zaključak je da nije dovoljno i da se treba još truditi. Ja sam nekako razmišljao o našem obrazovanju i kompjuterima. Interesantno je da Englezi nemaju onu izreku o bogu i šešir-džiji.

Upravo je u toku završno čišćenje raznih novinarskih redakcija. Od sad pa nadalje ništa više neće biti kao nekad. Trenutno u Velikoj Britaniji ne postoji nijedna značajnija novinska kuća koja svoju redakciju nije delimično ili potpuno kompjuterizovala. Gomile terminala po prostoriji, nema ono kuc-kuc-kuc-zvrnc, tekst na disketi, još dalje tekst direktno u urednikov deo memorije, a odatle pravo do terminala tehničkog urednika. Fascinantno i zastrašujuće.

Poslovna strana fino napreduje. Velika IBM-ova kula još uvek dominira dalom Londona kojim se dolazi od aerodroma Hitrou. Vrednost deonica velikih britanskih računarskih firmi (za većinu njih verovatno niste ni čuli) sasvim lepo napreduje) tako mi bar kažu oni koji se u to razumeju), a gradić Slough u blizini Londona ima više računarska industrija čak i od Slovenije. Da, bilo je prošle godine i krize. Neke firme su propale, ali kapital je ostao, kao i ljudi. Svi programeri koji su na ovaj ili onaj način učestvovali u velikom prodoru pre pola decenije i sad su u računarskom biznisu. Jednom rečju vitalno.

Peti korak ili od Vardara pa do Triglava

Sasvim je normalno da Jugoslovena interesuju Jugosloveni. Obilazio sam kompjuterske prodavnice i pored ostalih sitnih i nebitnih stvari pitao i da li su trgovali sa Jugoslovenima. To je lako saznati zato što svi kupci računara popunjavaju formular za oslobađanje od poreza na promet (VAT) zato što iznose računar iz Velike Britanije. Nisam imao mnogo sreće sa tim pitanjem. Većina prodavaca se ili ne seća nečeg značajnijeg vezanog za onih par Jugoslovena koji su kod njih kupovali, ili su suviše džentlmeni da bi rekli ako se sećaju. Zato se o toj temi raspričao Mark Windham, glavni prodavac najveće prodavnice kompjuterima u Redingu.

Da, kupovalo je nekoliko Jugoslovena kod njega (Reding je jedno od mesta gde u Britaniji živi veća koncentracija Jugoslovena). Da, dobro se seća. Šta mu je ostalo u pamćenju? To da ga je taj čovek zamolio da na računu napiše manju cenu od prave. To mu je bilo jako simpatično ali nije mogao da mu pomogne. Da li se seća još nečega? Da, načina kako mu je taj neidentifikovani Jugosloven prvo objasnio sve o kompjuteru koji namerava da kupi (dragi anonimni čitaoča da li se prepoznaješ?) i to nonšalancijom koja je sugerisala da mu je to jedino zanimanje u životu — da ide od kompjuterske radnje do kompjuterske radnje i objašnjava prodavcima sve o kompjuterima. Da li se seća šta je Jugosloven kupio. Čini mu se da je to bio neki „amstrad". Da li Jugosloveni često svraćaju? Da, nedavno je jedan kupio „amstrad PC"-je. Kaže da su svi Jugosloveni koji su kupovali džojstika insistirali na Quick Shotu II. Interesuje ga da otkud to. Takođe ga interesuje zašto kupac koji je upravo kupio neki računar pokupi uz to i kilograma prospekata svih ostalih računara. I još nekoliko sitnih pitanja. Svet se interesuje za nas.

Šesti korak ili Amstrad the Great

Ako želite da prodate vaš računar morate da ga napumpate toliko da bude veći od Big Bena i sjajniji od Sohoa. Ako želite da nastavite da ga prodajete morate sve to da duplirate ili da se potrudite da bude zaista dobar. Još uvek se ne zna koliko je „amstrad" PC zaista „dobar", ali je prilično jasno da je Šuger učinio sve što je mogao da pospeši prodaju. Od „amstrad PC"-ja se ne može pobeći. Novine su pune reklama na celoj strani. U većim robnim kućama možete da ga vidite na svakom ćošku. U odeljenju sa igračkama, među elektronskom robom, na školskom odeljenju, nema ga samo u opremi za kupatilo. Na televiziji se non-stop emituju kratke brze i efektne reklame, u kojima izgleda tako dobro da poželite da kupite bar tri četiri komada. Stručna štampa ne piše baš nešto previše o njemu, ali mu zato svi ostali posvećuju veliku pažnju. S ponosom se ističu prve količine prodate u Americi (to je sinrom Yugo ponosa). To su, praktično jedini računari koji se mogu videti baš u svim izlozima radnji sa elektronskom robom. Ogromni posteri objavljuju da je izašao još jedan — taj i taj — program, koji se može koristiti samo na „amstradu PC".

Da li treba još nešto da se doda? Vruća roba. Čudno, niko se ne izjašnjava o konkretnim detaljima računara. Valjda je taj računar otelotvorenje onog što se u računarskoj industriji, a i drugde, naziva kornim kompromisom.

Sedmi korak ili povratak bez zaključka

Zašto bez zaključka? Zato što je sve pametno o razlikama o onome što se mora i što hoćemo o tome kako bi trebalo o suptilnim analizama, već odavno rečeno a ovo je ionako bila samo šetnja sa pokojom tezom. Povratak izgleda ovako. Autobus koji nas vozi na aerodrom je opremljen malim kompjuterskim sistemom za izdavanje karata koji vozač kontroliše levom rukom. Neposredno ispred aerodroma je ogromna tabla (8x5 m) sa reklamom za računar taj i taj, malo dalje još jedan i tako dalje. Čekamo prtljag i ne mogu da ne primetim da je službenik, pored standardnog, opremljen i malim DEC-ovim terminalom, i da se na stolu ispred njega nalazi i Pelon Organiser II. Šta će mu sve to? Prolazimo carinu i ulazimo u tri šop. Na ponosnom mestu među elektronskom robom — „amstrad PC". Na tom mestu dobijem čudan i neodređen osećaj, odem do prodavnice novine, kupim oko tri kilograma kompjuterske štampe i vratim se kući. Fino putovanje. Nije bilo poučno ali je bilo prijatno. Ili tako nekako.

Branko Đaković

Rej Ouzi, autor programa Symphony

Raspevani softver Reja Ouzija

U svojoj izuzetno zanimljivoj biblioteci za Prave Programere „Microsoft Press" je nedavno objavio zbornik tematskih razgovora sa petnaest najbriljantnijih programera današnjice. U toku nekoliko narednih meseci, u okviru serije „Programeri govore", prenećemo najzanimljivije intervjue iz ove knjige — sa Garijem Kidalom autorom CP/M operativnog sistema, Džefom Raskinom, vođom projekta „mekintoš", Džonatanom Saksom, tvorcem „Loutsa 1-2-3", Bilom Gejtsom, tvorcem bejzika i mnogim drugim živim programerskim legendama. Ima li boljih učitelja da nas uvedu u svet programskih ideja, tajni programerskog zanata i tajni uspeha u programerskom poslu?

Diploma inženjera kompjuterskih nauka stečena na Univerzitetu u Ilinoisu i, već u početku, praksa kod „Dalta General" za mnoge programere bi bili ispunjeni snovi. Ambiciozni Rej Ouzi (Ray Ozzie, rođen 1955) smatrao je da je to samo dobar start i uporno nastavio da traži šansu za svoje stvaralaštvo. U pravom trenutku je prihvatio poziv Džonatana Saksa i Miča Kapora (o kojima smo pisali u pretprošlom broju) i u njihovom „Lotusu" izgradio program Symphony. Kad je postao „ime", osnovao je — kako to već biva u Americi — svoju kompaniju „Iris". Za razliku od nekih drugih vodećih programera, Ouzi živi u srećnom braku i u svojoj supruzi nalazi veliku podršku.

— Molim vas da za početak odgovorite na pitanje da li postoji neka formula za proizvodnju dobrih programa?

Za mene je to rigorozna struktura, krajnje dosledna i čista. Takođe verujem u visoko moduliran i slojevit softver. Ako ste prinuđeni da gradite komponente odvojeno, tada interfejsi više štrče, što znači da ih morate oblikovati.

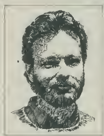
Kad više ljudi radi na jednom programu veoma je važno da se — u ranoj fazi projekta — utvrdi kako će se globalno postupati s greškama, prenositi argumenti i označavati potprogramske konvencije (bez obzira na to što se možda svako neće složiti s njima). Inače, nipošto ne treba govoriti drugima kako da komentarišu svoj kod, kako da koriste zagrade ili kako da nazubljuju programe. A kad radite u tuđem modulu, poželjno je da se koriste autorove konvencije. Tako se uči da se sarađuje s drugima.

Neophodna je klima povoljna za razmenu ideja. Na sastancima posvećenim razmatranju projekta dobro je da se čuje živa, čak burne diskusija. Iskustvo mi govori da su mnogi vrsni dizajneri veoma jogunasti i ne odriču se tako lako onog što smatraju da je ispravno. Ipak oni znaju kada i kako treba da odstupe.

— Kako vam je pala na um ideja za program na kome sada radite?

Mada još nije trenutak da otvorim sve karte o tekućem programu, mogu reći da je većina mojih produkata projektovana sukcesivnom aproksimacijom i usavršavanjem, a ne kao rezultat neće slučajne „ideje". Projekat počinje željom da se stvori neki proizvod koji će pomoći ljudima da u svojoj profesiji koriste računare efikasnije i sa većim zadovoljstvom.

Uočljiva je tendencija da se pišu programi samo radi novca, a ne da se rešavaju određeni problemi. Umesto da budu inovatori, pojedinci se upoznaju s postojećim i onda pokušavaju da



„Mislim da je prohujao poslednji talas kreatora koji su mogli da stvore nešto veliko u ovoj industriji."

kopiraju uveravajući sebe: „Hej, ja mogu da napravim nešto bolje od ovog!" Koliki su samo napori i novci straćeni u pronalaženju boljih spredšitova, kao da je svetu potreban spredšit u pedeset i sedam varijanti. Mislim da je prohujao poslednji talas kreatora koji su mogli da stvore nešto veliko u ovoj industriji. Sada kada su nestali snovi od sedam brojki, ljudi će morati da ozbiljnije prihvate vertikalna tržišta i razvijaju softver za specifične primene — pre nego da izmišljaju stvari koje nikome nisu potrebne.

Naš sadašnji produkt nije neki zamaskirani plagijat. Visoko produktivan, verujem da će biti koristan velikom broju ljudi.

— Kad oblikujete zamisao o nekom softverskom produktu, jeste li sigurni u sebe da ćete ga realizovati i isporučiti na vreme?

Da, prilično sam siguran. Imam dovoljno iskustva da znam šta mogu a šta ne mogu da uradim. Kompleksno programiranje projekta kojim rukovode menadžeri najčešće omane jer oni ne shvataju komponente projekta a još manje ličnost programera.

Što se mene tiče, ja se trudim da upoznam koliko god je to moguće porodičnu situaciju, način života i radne navike ljudi koji sa mnom sarađuju. I loše bi to bilo ako bi se sve vreme dizajniranja radili iz dana u dan od devet do pet; ja to od njih ne tražim, ali kad treba zapeti oni ne pitaju za radno vreme i na njih mogu da računam. Ali takođe treba osetiti trenutak kad im je neophodan predah.

Dajem prednost saradnicima koji su iskusni. Tražim od programera da sam predvidi vreme koje će mu biti potrebno da obavi određeni zadatak, pa onda svoje projekcije zasnivam na njegovim rokovima. Za programere-juniore karakteristično je da precenjuju svoje mogućnosti, a godinama, kada upoznaju zamke programiranja, postaju realniji, postojaniji i više motivisani u svom radu.

— Kad govorimo o poštovanju čvrstih rokova i realizovanju visoko kvalitetnih produkata, čini mi se da sve zavisi od menadžerskog stila?

Mnogi menadžeri smatraju da je s programerima teško raditi. Ja retko imam taj problem. Teškoće često iskrsavaju kad direktor izdaje stroge zapovesti ili sprovodi preteranu kontrolu. Programeri su veoma kreativna bića, visoko motivisana, sklona samokontroli. Ako to shvatite, onda ćete izbegavati da ih omatate administrativnim odlukama. Štaviše, olakšaćete život svom timu, oni će to ceniti i kasnije, kad to bude potrebno, višestruko se odužiti.

Ukupna atmosfera mnogo znači za uspešno programiranje. Kod nas to su lepe prostorije s najboljom opremom koju sebi možemo da dozvolimo, stereo u svakoj sobi i frižider koji je uvek pun. Neki saradnici „Irisovog" tima vole da rade kod kuće, tada im kompanija u stanu postavlja identičnu mašinsku garnituru.

— Maločas ste rekli da dajete prednost ljudima sa iskustvom. Šta pod tim podrazumevate?

Biti iskusan znači da ste radili na raznim poslovima. Za mlade ljude najbolje je da posle završetka koledža intenzivno rade na nekom poslu bar godinu dana, a zatim da se prebace u sasvim drukčiju oblast kompjuterske nauke. Na primer, počinjete sa operativnim sistemima, zatim prelazite na mrežu, grafiku, kompajlere ili bazu podataka. Gledano dugoročno, kao generalni programer dobićete bolju cenu za svoj rad. Za razliku od juniora, prekaljeni programer se

„Dobre izglede imaju programeri koji shvataju potrebe korisnika vertikalnog tržišta."

snalazi u svim poslovima, zna da svoj rad uskladi sa svojim sposobnostima i načinom života, u stanju je da apstraktno razmišlja, dobro sarađuje s drugima i uvek je visoko motivisan.

— Recite mi kako ste postali programer?

Sve je počelo 1969. na prvoj godini koledža. Nastavnik iz matematike pokazao nam je kalkulator za programiranje — bio je to Olivetti-Underwood Programma 101 — nastojeći da nas zainteresuje za tu novu igračku. Nas nekolicine

se odazvala pa smo zatim ispitivali sposobnosti tog kalkulatora. Saznali smo da nastavnik takođe ima neku svoju računarsku igračku. Naime, u matematičkom kabinetu se nalazio teleprinter uključen u mrežu školskog okruga, bio je to model General Electric 400. Naravno, mi najzagriženiji hteli smo da učestvujemo u akciji. Saznali smo šifru i počeli da se lepo zabavljamo programirajući u bejziku i fortranu. Konačno, zasitio sam se te igre, pogotovu što su me privukle druge aktivnosti.

— Koje vrste?

Zainteresovao sam se za elektroniku, pa sam se upisao na kurs za elektronske tehničare. Ubrzo sam shvatio da to nije ona prava stvar. Voleo sam da petljam sa elektronikom, ali me matematička teorija nije privlačila. Tada me je jedan poznanik nagovorio da se upišem na kurs za programere. Zagrizao sam udicu iz prve, mašinski problemi su bili laki, gotovo zabavni. Programiranje sam zavoleo iz istog razloga kao i elektroniku: mogao sam da pravim male stvari, ovog puta programe umesto elektronskih sprava.

— Znači, bili ste upecani?

Udica — kompjuterski sistem PLATO — nalazila se u univerzitetskom kampusu. Da bih dobio pravo pristupa sistemu, upisao sam se na kurs i to je bio stvarni početak mog programerskog iskustva. Računar mi je postao navika. Svake noći posle deset mašina je bila slobodna pa je grupica zanesenjaka, među kojima sam bio i ja, radila sve do šest ujutru. Ta moja hakerska faza potrajala je nekoliko godina.

Od trenutka kad mi je omogućen pristup računaru, redovne školske obaveze su otišle u drugi plan. U stvari, nisam prekinuo veze sa školom samo zato da ne bih izgubio pravo da koristim računar. Naravno, nisam bio zapažen po uspehu i trebalo mi je pet i po godina da završim koledž.

„Velika je stvar kad pomislite da neki čovek na drugom kraju zemlje koristi vaš softver i, veoma zadovoljan, oseća potrebu da vam pošalje pismo."

— Kako ste premostili jaz između studentskog i sistemskog programiranja?

Zahvaljujući nekim poukama još iz vremena kad sam koristio PLATO. Posle devetomesečne borbe da dobijem pravo pristupa, shvatio sam da nije dovoljno biti dobar u programiranju i raditi taj posao punim srcem. Morali ste da budete vešt manipulator poput političara. Postojala je stroga hijerarhija privilegija za korišćenje PLATO-sistema. U stvari, bio je to kompjuterizovani nastavni sistem uključen u masivnu strukturu „Control Data" sa gotovo hiljadu terminala širom sveta. PLATO je projektovala i realizovala jedna malobrojna zatvorena grupa ljudi, oni su bili veoma posesivni ne samo u odnosu na svoj izum, već i kad je trebalo nekome dozvoliti da sistemski programira.

U početku tu privilegiju su imali samo članovi fakulteta, učesnici u projektovanju i nekoliko studenata koji su se ubacili u unutrašnji krug. Svi ostali koji su hteli da programiraju mogli su da koriste „Tutor" sa interpretivnim jezikom srednjeg nivoa. Pošto sam čvrsto odlučio da se uključim u sistemsko programiranje, konačno sam izmanipulisao da budem primljen u povlašćeni krug.

— Gde ste se najpre zaposlili?

Razmišljao sam o poslu prilično zabrinuto još pre nego što sam diplomirao. Srećom, te 1977. godinu svuda su tražili programere, bio sam intervjuisan u dvanaest kompanija i svuda sam pozitivno ocenjen, uglavnom zbog mog praktičnog iskustva. Konačno sam se zaposlio kod „Data General" i počeo da radim za Džonatana Saksa. Trebalo je da projektujemo jedan novi operativni sistem i arhitekturu počevši od nule. Jezgro radna grupa imalo je samo tri čoveka.

— Koliko dugo ste se zadržali kod „Data General"?

Još godinu dana. Počeo sam da se frustriram u velikim kompanijama teško ostvarujete svoje ciljeve. Želeo sam da radim za neku malu kompaniju čiji program uključuje mikrokompjutere. Jedna žena, moj posrednik, uporno je tragala za takvom prilikom i, gle, posle godinu dana mi je javila da je našla ono što me interesuje. Bila je to kompanija „Software Arts". Intervjuisali su me i primili kao službenika, 29-og po redu.

— Kako ste se snašli kod „Software Arts"-a?

Moj prvi posao je bio Radio Shack TRS-80 Model 3. Radio sam na jezičkom interpreteru postavljajući podlogu za primenu TK! Solvera. Nije prošlo ni mesec dana kako sam stigao a

„Programeri su veoma kreativna bića, visoko motivisana i sklona samokontroli. Ako menadžeri to shvate, onda će izbegavati da ih ometaju administrativnim odlukama."

kompanija je kupila novu mašinu za razvoj VisiCalc-IBM PC. Pošto su mi dozvolili da eksperimentišem nekoliko dana na novoj mašini, shvatio sam da je to ono što odavno priželjkujem.

— Da li su i drugi programeri imali tu želju?

Apsolutno. U to vreme nisam mnogo cenio tržište. U duši sam još uvek bio haker i razmišljao sam: IBM s mikroračunarom, to znači da mnogo šta otpada, cene će se smanjiti i računari će postati dostupniji širem krugu. Programeri kao ja ne bi više bili zavisni od glomaznih struktura.

IBM PC bio je prvi mikroračunar dovoljno moćan da kompajlira jedan program na kome sam upravo radio. Mogao sam da uređujem tekst i kompajliram program, povezujem ga i proveravam. A kad je zatim stigao prvi hard-disk, to je bilo kao da je san postao java. Satima bih sedeo u svom uredu i radio na računaru koliko mi srce želi. Dokumentacija je bila užasna, ali ja o tome nisam hajao — od silne sreće.

— Izgleda mi da ste bili prilično zadovoljni u „Software Arts"-u. Da li ste pomišljali da promenite sredinu?

Jednog dana me je Džonatan Saks pozvao na ručak. Bio je napustio „Concentric Data" i pridružio se Miču Kaporu koji je osnovao malu kompaniju „Micro Finance Systems". Interesovao se da li bih došao kod njih, a ja sam ga pitao na čemu rade. Ostavio me je u utisku da rade na jednom klonu VisiCalc-a. Kod „Software Arts"-a se razvijao stvarni VisiCalc i zašto bih — razmišljao sam — sada prelazio u drugu kompaniju da radim na imitaciji. Odlučio sam da pričekam. U međuvremenu, Džonatan i Mič su izgradili Lotus 1-2-3.

— Kad ste ponovo videli Saksa?

Ponovo smo se sreli — tada sam upoznao i Miča Kapora — na izložbi COMDEX, kada je javno predstavljen Lotus 1-2-3. U to vreme sam imao staž od dvadeset meseci kod „Software Arts"-a. TK! Solver je bio savršen i već lansiran. Džonatan i ja smo dugo razgovarali. Mada je 1-2-3 sada na tržištu, objašnjavao mi je, on postoji još duga lista novih ideja kojima on i Mič žele da obogate drugu verziju. Ponovo me je pozvao i ja sam obećao da ću ozbiljno razmisliti.

— Dakle, bili ste spremni da se rastanete od „Software Arts"-a?

Posle razgovora sa Mičom Kaporom pristao sam da pređem u „Lotus" i radim na jednom specifičnom programu — to će kasnije postati Symphony.

— Kako je izgledao rad u „Lotusu"?

Pa, od prvih dana sam konstatovao da mi od kuće do sedišta kompanije u Kembridžu potreban sat vožnje. Zato sam se dogovorio sa Mičom da „Lotus" otvori mali ured u Litltonu koji je bio mnogo bliži meni, Džonatanu i Bariju Spenseru. Nas trojica smo bili zaduženi za novi program.

— Da li su osnovne zamisli o Symphony već postojale kada ste se priključili timu?

Postojali su samo jedan uopšten koncept na pet-šest strana i beskrajna lista karakteristika koje bi trebalo, ako je to ikako moguće, da budu uključene u poduhvat.

Počeo sam da radim na glavnom segmentu, procesoru reči. Bari Spenser je bio odgovoran za komunikacije. Ali Džonatan se posle nekoliko meseci svega zasitio, namučila ga je jedna komponenta koda na kojoj je dugo i naporno radio. Povukao se i tako sam ja postao nosilac celog projekta. Da bi nas bilo bar trojica u timu, angažovali smo Meta Sterna, veoma dobrog programera. Tokom sledećih devet meseci radili smo kao ludi. Ja bih odlazio u „Lotus" svake druge nedelje i obaveštavao Miča o napredovanju. Procenio bi ono što smo uradili, eventualno uskladio nove rokove i mi bismo nastavili.

— Dakle, bili ste veoma efikasni?

Apsolutno. Mič Kapor je jedan od razloga zašto je „Lotus" najbolja kompanija za koju sam ikad radio. Nije razmišljao o tome da li neko ugrožava njegov autoritet. Interesovao se mnogo više za to kako će izgledati korisnikov interfejs, recimo, dok je algoritmima poklanjao manju pažnju. Sve u svemu, kontrolu nad razvojem prepustio je nama.

— Kako je jedan složen program kao Symphony mogao da bude realizovan tako lako?

Jedan od razloga je bio u tome što je kompanija glavninu svojih napora usmerila na Symphony. Svako je bio ponosan na produkt pa je nastojao da svoj posao završi na vreme. Uz to, sa zebnjom se očekivalo šta će uraditi potencijalni konkurenti — Ovation i Framework.

Drugi razlog je svakako u dobrom međusobnom komuniciranju. Mali tim u izolovanoj sredini komunicira veoma efikasno. Konačno, naš tim je bio malobrojan. Smatram da na projektu realizacije produkta ne treba da radi više od pet ljudi, izuzimajući naravno masovne sisteme kao onaj koji evidentira i zadužuje poreske obveznike u celoj zemlji.

— Zašto mislite da je mali tim efikasniji od velikog?

U timu sa više od pet ljudi komuniciranje se otežava do te mere da nastaju problemi u kvalitetu produkta. Tako, bagovi se često javljaju kad je loš interfejs između podsistema, a to je najčešće rezultat neadekvatnog komuniciranja. Kad se bag otkrije, tendencija je da se ispravka vrši u podsistemu, a ne gleda se program kao celina. Ukoliko nemate par ljudi koji poznaju funkcionisanje svakog podsistema i kako se on uklapa u celinu, svi su izgledi da će ceo produkt biti manjkav.

„Računar je jedna velika, širom otvorena kutija sa alatom."

— Koji su drugi važni kriterijumi za dobar program?

Važno je da imate veoma čist konzistentan arhitektonski model produkta dok radite na njemu. Naravno, u hodu možete da menjate njegovu arhitekturu ali [illegible] arhitektonski sklop na kraju lovljenjem bagova, te je loš posao. Ispostavi se da umesto dva-tri minorna baga morate da otklanjate duboke pukotine.

Lotus 1-2-3 je lepo projektovan i konzistentan program. Na tom čvrstom produktu je izgrađen Symphony — veoma opsežan program s mnogim interakcijama između komponenata. Nikad nije imao neke ozbiljne bagove i za poznavaoce nema tajni oko interakcija između podsistema bazičnog 1-2-3.

— Kad je Symphony bio završen, šta ste radili?

Mič je održao svoja obećanja i ja sam mogao da se posvetim specifikacijama koje sam započeo pre „Lotusa". Posao sam uspešno završio i posle temeljitog proveravanja „Lotus" se saglasio da finansira moj projekt kroz posebnu kompaniju — „Iris Associates". Odmah sam u „Iris" pozvao Tima Halvorsona i Lena Kavena, poznanika iz „Digitala", a kasnije smo angažovali još nekoliko ljudi.

—REJ OUZI—

— Šta mislite kakvi će programi biti na tržištu kroz pet ili deset godina?

To bih i ja voleo da znam! Kao mnogi drugi pogrešno sam verovao da postoji tržište kućnih računara. Takođe ranije sam mislio da bi lični taym-menadžment program bio veoma koristan, ispostavilo se da je mnogo jednostavnije da zavirim u notes nego da iz torbe vadim računar, završavam ga i priključujem da bih saznao razlo koje sam sastanka zakazao za sledeći dan.

— Da li imete neki poseban savet za one koji žele da razvijaju softver?

Dobre izglede imaju programeri koji shvataju potrebe korisnika vertikalnog tržišta. Kad je reč o horizontalnim aplikacijama za masovno tržište može da postoji samo ograničen broj programa kao što je 1-2-3. Verovatno ih možete na prste izbrojati: to su procesori reči, spredšiti, baze podataka. Ako personalni računari postignu uspeh — dugoročno gledano — to će biti zbog toga što će se programi krojiti tako da maksimalno zadovolje specifične korisnikove potrebe. Mislim da bi budući programeri trebalo manje da razmišljaju o Mesecu a veću pažnju da poklone nalaženju gnezda na zemlji.

— Kako uspevate da isporučite krajnjem korisniku baš ono što on želi?

Najpre nikad ne gubim iz vida da je osnovni cilj razvojno-proizvodnog ciklusa da se kupcu pruži doista nešto korisno. Zatim nastojim da bar približno procenim koji će procenat korisnika upotrebljavati ovu ili onu komponentu, što je značajan indikator kod projektovanja. Ako postoji neka opskurna funkcija koje će nešto značiti samo malom broju korisnika, ne ulažem veliki napor za njeno usavršavanje. Naravno, tek kad produkt isporučimo saznajemo da li smo tačno procenili profil kupca.

— Da li vam se korisnici javljaju?

O da. O tome bih mogao mnogo da vam pričam. Velika je stvar kad pomislite da neki čovek na drugom kraju zemlje koristi vaš softver i veoma zadovoljan oseće potrebu da vam piše. Ali nije prijatno kad vam neko javio da je otkrio bag u nekoj zabitoj komponenti.

Moje najčudnije iskustvo vezano je za jeden telefonski poziv baš u vreme kad je osnovan ... Javio mi se hirurg koji je koristio Symphony za analizu konkretnih podataka tokom operacije na otvorenom srcu. Zadrhtao sam pri pomisli da nekom srčanom bolesniku dok leži na operacionom stolu život zavisi od ispravnosti mog programa.

— Kad pišete program, da li je to za vas mukotrpan ili prijatan posao?

Ne bih se bavio tim poslom da mi ne pričinjava zadovoljstvo. Neprijatno mi je samo kad u vreme ciklusa intenzivnog razvoja ne stižem da vidim svoju suprugu Danu i sine, ona mi pruža veliku podršku.

— Šta vas fascinira u programiranju?

Programiranje je oblast za one koji su skloni kombinovanju. Ali kombinatoru su potrebna oruđa. Elektro-inženjeri raspolažu raznim komponentama koje kombinuju da bi nešto izgradili, međutim ograničeni su time što ne raspolažu fizičkom opremom. Sa računarima — ako nešto smislite to možete i da uradite. Projektujete sopstvene alatke i stvarate potrebne segmente toka rada. Ako vam se nešto ne sviđa to možete da popravite, izmenite ili odbacite. Računar vam je jedna velika širom otvorena kutija sa alatom. Jedini ograničavajući faktor je količina vremena koje je potrebno mašini da uradi zadatak, kao i vreme koje je vama potrebno da napišete program.

— Šta biste savetovali mladim programerima?

Ako pripadate ljudima koji su fascinirani programiranjem ja bih vas savetovao da budete uporni i da programirate što više i što različitije projekte. Provedite za računarom maksimalno moguće vreme, ali naučite da procenjujete i poštujete granice svojih fizičkih i psihičkih snaga, i ne sekirajte se ako vas ljudi smatraju čudakom.

Interfejsi za muzičke instrumente

Ne pucaj u pijanistu

Odavno su prohujala vremena kada su Betoven i Mocart provodili sate i sate pišući svoje kompozicije i mukotrpno bilježili svaki ton koji im je izgledao dobar. Savremeni kompozitori i muzičari-amateri imaju u (kućnim) računarima značajnog pomoćnika. Gotovo svaki kućni računar može se opremiti interfejsom za povezivanje sa elektronskim muzičkim instrumentima, a kod novijih modela takav interfejs predstavlja deo standardne opreme.



MIDI je skraćenica od Musical Instrumente Digital Interface. To je serijski interfejs nalik na RS232 koji omogućava povezivanja računara sa maksimalno 16 elektronskih muzičkih instrumenata (sintesajzera, sekvencera, ritam mašina itd.) ili međusobno povezivanje elektronskih instrumenata. Druga mogućnost je više teme nekog od muzičkih časopisa, te ćemo ovaj tekst posvetiti povezivanju muzičkih instrumenata sa računarom.

MIDI interfejse možemo podijeliti na jednostrane i dvostrane. Jednostrani MIDI omogućava samo prenos podataka iz računara na instrument, dok dvostrani MIDI omogućava i obratan prenos, tj. iz instrumenta u računar. Jednostrani MIDI dovoljan je za većinu primjena, ali dvostrani MIDI ni u kom slučaju nije na odmet.

MIDI nije čest gost na kućnim računarima, mada se može nabaviti u vidu dodatnog hardvera za gotovo svaki poznati računar (uključujući i „spektrum" i „komodor"). „SPEKTRUM 128" i +2 sa jednostranim i „atari ST" računari sa dvostranim MIDI predstavljaju među kućnim računarima retka izuzetke koji su opremljeni ovim interfejsom.

Sa ili bez softvera

Na većini računara ne postoje instrukcije za kontrolu MIDI-ja, pa čak ni na „starlju" koji ga ima ugrađenog. Primjena MIDI interfejsa u ovakvim slučajevima uglavnom je ograničena na softver (koji je gotovo nemoguće nabaviti na domaćem tržištu) ili korištenja OUT ili POKE instrukcija. Jedan od rijetkih izuzetaka je „spektrum 128" koji posjeduje komanda za rad sa MIDI-jem u ROM-u. Nažalost, njegov MIDI je jednostranog tipa.

Zamislimo da želimo da nam instrument odsvira neki ton, recimo ton g u oktavi 1 (tj. g1). Jedino što je potrebno to je da pošaljemo NOTE ON EVENT informaciju. Ona se sastoji od 3 bajta. Prvi bajt se računa po formuli 143+ch. Šta je sada ch ??? Pošto preko MIDI-ja možemo da kontrolišemo više instrumenata, uveden je takozvani MIDI kanal. Svakom od priključenih instrumenata možemo (i moramo) dodijeliti određeni kanal (broj od 1 do 16). Ukoliko smo, na primjer, nekom sintesajzeru dodijelili kanal 5, on će primati samo one informacije koje se odnose na kanal 5 i upravo je taj kanal parametar „ch" u maloprije pomenutoj formuli. Pošto ćemo najčešće koristiti samo jedan instrument, najbolje je uzeti ch=1.

Drugi bajt određuje visinu tona na već opisani način. U našem slučaju za ton g1 to će biti 67.

Treći bajt određuje glasnoću (1 najtiše, 127 najglasnije). Ovaj podatak mora se poslati čak i ako ga instrument ne podržava (u tom slučaju biće ignorisan). Najbolje je uzeti 64 (prosječna glasnoća).

Dakle, da bismo proizveli ton g1 na instrumentu 1 prosječnom glasnoćom, treba da pošaljemo bajtove 144,67 i 64. Ali, kako ćemo ih poslati??? Obično preko OUT instrukcija koje ovise kako od računara, tako i od firme koja je proizvela interfejs

„Ateri" koristi port 3 dok najrasprostranjeniji MIDI interfejs za „spektrum 48" i „komodor 64" koriste portove 191 odnosno 56837. Dakle, na „atariju" ćemo izvršiti OUT 3,144 OUT 3,67 OUT 3,64 na „spektrumu" OUT 191,144 OUT 191,67 OUT 191,64 a na „komodoru" POKE 56837,144 POKE 56837,67 POKE 56837,64. Ne treba zaboraviti da se kod „spektruma" i „komodora" interfejs treba prethodno inicijalizirati pomoću OUT 159,3 OUT 159,66 odnosno POKE 56838,3 POKE 56836,66

Na ovaj način smo proizveli ton. Ali do kad će on svirati? Sve dok ne pošaljemo NOTE OFF EVENT informaciju. Ona je slična NOTE ON EVENT informaciji samo što je sada prvi bajt 127+ch, a treći nema nikakvu ulogu

Pod strogom kontrolom

Već i ovo što je rečeno omogućava vam da pišete simfonije ali ovim mogućnosti MIDI-ja nisu ni djelimično iskorištene. Pomenućemo neke od najznačajnijih stvari. Kodovi za te „sitnice" nisu u potpunosti standardizovani pa se može desiti da ne rade na svim instrumentima. Kodovi koji budu pomenuti isprobani su na nekoliko sintesajzera firme KORG

Mnogi instrumenti posjeduju modulator koji omogućava razne vrste modulacija, recimo frekventnu (DCO — vibrato) ili „cutoff" (VCF — „wahwah" efekat) modulaciju. Prvu možete postići pomoću niza bajtova 175+ch,1,m a drugu pomoću 175+ch,2,m, gdje parametar m određuje stepen modulacije (0—127). Ostale vrste modulacija možete postići mijenjajući drugi bajt

Boju zvuka možete mijenjati pomoću sekvence 191+ch,n gdje je n broj „programa" na instrumentu u kojem je smještena željena boja tona

Sekvence 125,0 prebacuje instrument u takozvani OMNI mod u kojem instrument prima podatke bez obzira na koji su kanal upućeni. Usput, ovo je ujedno i najlakši način da prekinemo izvođenje svih tonova odjednom. Ovo stanje se poništava sa 175+ch,124,0

Postoje i kodovi za kontrolu rada sekvencera (naprava koje omogućuju memorisanje melodije koja je svirana na instrumentu). Kod 250 aktivira sekvencer, kod 252 ga deaktivira, dok 248 šalje CLOCK impuls koji sekvencer shvata kao zahtjev da treba odsvirati slijedeći ton. Ovi kodovi spadaju u grupu SYSTEM REAL TIME MESSAGES

Većina elektronskih muzičkih instrumenata posjeduje interni RAM u koji se pohranjuju podaci o bojama tonova (talasni oblici, envelope, itd.) i još neki podaci. Pomoću određenih kodova (koji strogo zavise od instrumenta) možemo da pošaljemo neki podatak u određenu lokaciju RAM-a instrumenta, ili da postavimo zahtjev da nam instrument dostavi sadržaj određene memorijske lokacije (za ovo drugo moramo imati dvostrani MIDI)

U oba smere

Dvostrani MIDI omogućava i komunikaciju instrument-računar. Naime dok na instrumentu sviramo neku melodiju, itd. instrument šalje računaru NOTE ON, NOTE OFF EVENT informaciju ili neke informacije druge prirode koje možemo na računaru čitati pomoću IN ili PEEK. Na ovaj način, pomoću računara možemo memorisati melodiju koju smo svirali i kasnije je reprodukovati

Da bismo saznali sadržaj nekog bajta u RAM-u instrumenta, moramo poslati specijalni kod (tzv. DATA REQUEST) pa će instrument po prijemu ovog zahtjeva poslati datu vrijednost.

Pošto je prenos bajtova preko MIDI-ja vrlo brz (31250 boda), podaci iz (ili u) RAM-a instrumenta obično se šalju u vidu blokova (npr. po 1K) sa kontrolnim bajtom na kraju. Ukoliko instrument detektuje grešku u prijemu podataka, šalje određene bajtove računaru kao znak da je nastupila greška

MIDI na „spektrumu"

MIDI na „spektrumu 128" i +2 je podržan bejzik komandom PLAY u potpuno istom obliku kao i kod kontrole zvučnog čipa (vidi računare 16), s tom razlikom što sada možemo koristiti svih osam kanala. Svaki string koji želimo poslati na MIDI moramo započeti oznakom „Y" iza čega slijedi broj MIDI kanala (npr. PLAY „Y1cdefgabC"). Ukoliko se ova oznaka nađe u jednom od prva tri stringa, taj string će biti i pod kontrolom zvučnog čipa i MIDI-ja. Ovo možemo izbjeći stavljanjem tri prazna stringa na početak (npr. PLAY "", "", "", "Y1cdefgabC"). Naravno možemo kombinovati tako da nam neke glasove svira zvučni čip, a neke MIDI

Unutar PLAY naredbe može se koristiti i oznaka „Z". Iza nje slijedi neki kod koji treba prenijeti na MIDI. Npr. PLAY „Z124Z0Y1c" prebacuje instrument u OMNI mod, a zatim izvodi ton „c"

Oznake W X U i M unutar instrukcije PLAY nemaju nikakvu ulogu pri kontroli MIDI-ja, te će biti ignorisane. Svi ostali znakovi imaće isto dejstvo kao kod zvučnog čipa (o čemu su „Računari" opširno pisali u broju 18 — „Dobro jutro, džezeri").

Jedini problem vezan za MIDI na 128-ici nastaje zbog nestandardnog priključka u Sinkler stilu. Čak i kad nabavimo odgovarajući utikač (uz nešto sreće možemo ga naći kod vlasnika QL-a) postavlja se pitanje kako napraviti kabl pošto nema nikakve dokumentacije o rasporedu pinova. Autor je metodom „probaj i vidi" pronašao da treba spojiti prvi i peti pin na računaru (gledano s desna nalijevo) sa drugim i četvrtim pinom na petopolnom ulazu na instrumentu (gledano u smjeru suprotno od kazaljke na satu)

MIDI iz mašinca

Na „spektrumu 48", „komodoru 64" „atariju" nema neke bitne razlike u korišćenju MIDI-ja iz bejzika i iz mašinca. Olakšavajuća je činjenica da se kod ovih računara bajtovi iz kompjutera šalju na interfejs paralelno (cijeli bajt odjednom), a interfejs ih razlaže na bitove i tako šalje dalje. No kod „spektruma 128" i +2 nije tako. Zbog toga je i uvedena subkomanda „Z" (koju ne možemo simulirati pomoću jednog OUT-a). Iz mašinca nam može poslužiti rutina u ROM-u 0 na adresi 286 koja šalje bajt iz akumulatora na MIDI (prije poziva ove rutine potrebno je isključiti interapt)

Ovim bi izlaganje o MIDI komunikaciji me bilo završeno. Ljubiteljima računara i elektronske muzike koji posjeduju neki od instrumenata sa MIDI priključkom ostaje još samo da svoj računar opreme interfejsom koji se može nabaviti po cijeni koja je slična cijeni RS232 interfejsa

Željko Jurić

MIDI informacije

CHANNEL VOICE MESSAGES

Ove informacije govore instrumentu koje tonove da svire, kakvom bojom tona, kojom glasnoćom, te kada će početi a kada završiti neki ton. Mogu se svrstati u nekoliko podgrupa

a) NOTE ON EVENT — informacija koja govori instrumentu da treba da započne određeni ton

b) NOTE OFF EVENT — informacija koja govori instrumentu da treba da prekine sa izvođenjem tona

c) CONTROL CHANGES — informacija koja služi za kontrolu efekata modulacije tona (vibrato itd.) koje podržavaju neki instrumenti

d) PROGRAM CHANGES — ova informacija određuje kojom će se bojom tona izvoditi pojedini tonovi. Naime, većina instrumenata omogućava samostalno kreiranje boje tona promjenom izvjesnih parametara (talasni oblik, envelope itd.). Kreirani zvukovi mogu da se smjeste negdje u memoriju instrumenta. Kasnije komandom „PROGRAM CHANGES" biramo jedan od kreiranih zvukova

e) CHANNEL PRESURE — ova se informacija rijetko koristi i služi za kontrolu nekih specijalnih efekata koje posjeduju skuplji sistemi

f) PITCH BENDER CHANGES — informacija koja se koristi za „štimanje" instrumenta, promjene tonaliteta, itd. Informacije pod a) i b) su složene i nose u sebi dva numerička podatka

a) NOTE NUMBER — ovaj podatak određuje visinu tona koji treba da se odsvira. Može biti u opsegu od 0 do 127. Brojka 60 predstavlja notu c1, 48 malo c, 61 cis1 62 d1 72 c2, itd. Ovaj podatak mijenja visinu tona u polustepenskim koracima i identičan je drugom parametru BEEP naredbe kod „spektruma" uvećan za 60

b) KEY VELOCITY — iako riječ velocity znači brzina, ovaj podatak nema nikakve veze sa brzinom već određuje kojom glasnoćom će se izvoditi određeni ton. Mnogi instrumenti prosto ignorišu ovaj podatak.

CHANNEL MODE MESSAGES

Služe za prenos podataka kojima se uključuju ili isključuju neki od režima rada instrumenta (OMNI POLY, MONO itd.)

SYSTEM MESSAGES

Ove informacije prenose parametre koji kontrolišu MIDI sistem. Mogu se podijeliti u 3 podgrupe: REAL TIME MESSAGES koje se koriste za kontrolu sekvencera i ritam mašina, COMMON MESSAGES koje se koriste samo na skupim sistemima za izbor melodije koja će biti korištena i EXCLUSIVE MESSAGES koje služe za komunikaciju između RAM memorije računara i RAM memorije instrumenta

Dejanove pitalice

Gomila jednostavnih brojeva

Znate li onu priču o ljudima kojima je rečeno da u mraku skupljaju kamenja preko koga prelaze, kako će se docnije kajati i onaj ko ponese malo i onaj ko ponese mnogo (posle se ispostavilo da su kamenčići bili dijamanti)? Tako je nekako i sa nama — ako damo tešku pitalicu, žao nam je što smo primili malo odgovora. Ukoliko, sa druge strane, damo laku pitalicu poput devetnaeste, kukamo što za tri dana treba da pregledamo nepojmljivu gomilu pisama! Devetnaesta pitalica će, dakle, ostati u sećanju kao najrešavanija — u predviđenom smo roku primili preko 500 (zapravo 501) pisama, među kojima smo pronašli 404 tačna odgovora.

Podsećanje na problem ovoga puta može da bude sasvim kratko. Posmatrajmo niz brojeva 5, 52, 525, 5252, 52525, ... — sledeći broj očito dobijamo kada prethodnom naizmenično dopisujemo dvojku, odnosno peticu. Trebalo je pronaći najmanji element niza koji je deljiv sa 99 ili (što je sasvim isto) reći koliko cifara taj broj ima.

U osnovnoj školi se uče pravila deljivosti: broj je na primer deljiv sa 9 ako i samo ako mu je zbir cifara deljiv sa 9. No, u školi verovatno niste naučili kada je broj deljiv sa 99 — potreban i dovoljan uslov je da broj bude deljiv sa 11 i sa 9. Postoji prilično zgodno pravilo kojim se određuje da li je broj deljiv sa 11: saberete sve cifre na parnim mestima (nazovimo ovaj zbir S1) i sve cifre na *neparnima* (S2). Broj je deljiv sa 11 ako i samo ako je S1—S2 deljivo sa 11.

Pre nego što se upustimo u matematičko rešavanje problema, pokušaćemo sa računarom. Slika 1 prikazuje bejzik program koji praktično trenutno proizvodi rešenje — broj 159. Kako ovaj program radi? Promenljiva N označava broj cifara, dok R označava ostatak pri deljenju sa 99: u početku je N=1 (broj je 5) i R=5 (broj 5 daje ostatak 5 kada pokušamo da ga delimo sa 99). Program povećava N i primenom jednostavne formule proračunava novi ostatak, posao se ponavlja sve dok je R različito od nule. Zadatak se primenom računara rešava i na nekoliko drugačijih načina, od kojih većina veoma brzo dovodi do korektnog odgovora.

Matematičko rešenje će nam oduzeti jedva malo više prostora. Pretpostavimo najpre da je broj cifara N paran, tada je zbir cifara broja An jednak *7*N/2*, dok je S1—S2 jednako *3*N/2*. Da bi broj bio deljiv sa 99 mora *7*N/2* biti deljivo sa 9, a *3*N/2* sa 11, ovaj uslov zadovoljavaju svi elementi An za koje je *N=99*K*. Da li se, prema tome, najmanji broj koji zadovoljava uslove sastoji od 99 cifara? Ne, jer je broj 99 neparan, a početna je pretpostavka da je N paran. Već za N=99*2=198 imamo jedno korektno rešenje koje smo, izgred budi rečeno, primili od desetak čitalaca.

slika 1:

```
10 [illegible]
20 T=TIME
30 N=N+1
40 [illegible]
50 [illegible]
60 IF R<>0 THEN GOTO 30
70 T1=TIME
80 PRINT "Broj ima: ";N;" cifara "
90 PRINT "Vreme izvrsavanja: ";
100 PRINT [illegible];" sekundi "
110 END

> RUN

Broj ima: [illegible] cifara. Vreme izvrsavanja: [illegible] sekundi
```

slika 2:

```
0 = 44 - 44
1 = 4 - 4 + 4 / 4
2 = 4 / 4 + 4 / 4
3 = (4 * 4 - 4) / 4
4 = 4! - 4*4 - 4
5 = 4 + (4-4)!
6 = 4! / (4 + 4 - 4)
7 = 4! / 4 + 4 / 4
8 = 4 + 4 - 4 + 4
9 = 4 + 4 + 4 / 4
10 = (4 - 4 / 4)! + 4
```

Razmotrimo situaciju kada je broj cifara neparan. Zbir cifara je tada *5*(N+1)/2*, dok je S1—S2 jednako *(3*N+3)/2*. Da bi prvi od ovih brojeva bio deljiv sa 9, a drugi sa 11, N treba da daje ostatak 6 pri deljenju sa 9 i ostatak 1 pri deljenju sa 11, N je uz to paran broj. Uz malo žongliranja sa formulama dobija se da je najmanje N koje zadovoljava sva tri uslova broj 159, kako je 159 manje od 198, traženi element niza ima 159 cifara.

Pošto smo primili mnogo relativno ravnopravnih rešenja, izdvojili smo desetak najbolje obrazloženih i među njima izvukli prvu nagradu — 20 000 dinara pripada *Stevanu Bogdanoviću* iz Majdanpeka. Zatim smo pomešali svih četiri stotine tačnih rešenja i izvukli drugu i treću nagradu — 15 000 dinara je dobio *Nedim Čolić* iz Sarajeva a 8 000 dinara *Ljubomir Josifovski* iz Skopja. Pohvale zaslužuju *Zoran Troter, Ivo Strmčević, Juraj Mucko, Miroslav Mitić* i *Nenad Grujić*.

Priču o devetnaestoj, do sada rekordnoj pitalici treba da završimo odavanjem priznanja *Branku Turku* iz Beograda koji je zamislio ovaj zanimljivi problem. Ovo je istovremeno i izazov za vas — ukoliko sastavite neki originalan i umereno težak zadatak koji se elegantno rešava primenom računara, setite se Dejanovih pitalica!

Po četiri četvorke

Dvadeset prva pitalica očekuje od vas da uz pomoć četiri četvorke i simbola matematičkih operacija predstavite što više uzastopnih prirodnih brojeva. Na slici 2 smo na primer predstavili brojeve između 0 i 10 pomoću četiri četvorke, u definicijama ne sme da se koristi ni jedna druga konstanta osim broja 4 (ne smete na primer da kvadrirate broj tako što ćete napisati 4^2 — tu se javlja i konstanta 2), dok je dijapazon matematičkih operatora približno prilagođen bejziku — zamislite da je Microsoftovom bejziku pridodat uzvičnik koji izračunava faktorijel, dok su mu oduzete trigonometrijske i logaritamske funkcije kao i konstanta Pi.

Od vas očekujemo da nastavite sliku 2 — predstavite broj 11, pa broj 12 i tako dalje. Kada vam posao dosadi, upišite najveći broj koji ste predstavili sa četiri četvorke u naš kupon (pažnja: ako napišete da ste predstavili broj 100, morate da nam pošaljete i formule za prvih 99 brojeva!) i pošaljite rešenje na adresu „Računari" (za Dejanove pitalice), Bulevar vojvode Mišića 17, Beograd, tako da pristigne pre 25. jula 1987. Sva pisma sa korektnim odgovorima konkurišu za novčane nagrade od 20 000, 15 000 i 8 000 dinara, dok će kuponi na koje je upisan identifikacioni broj učestvovati u godišnjem takmičenju rešavača pitalica. Identifikacioni broj dobijate tako što u prva tri kućice upišete ***poslednje*** *tri cifre nekog broja telefona, u sledeće dve godinu vašeg rođenja i na kraj dodate dve cifre po izboru.*

Problem, kao i obično, možete da rešavate „peške" i primenom kompjutera. „Pešačko" rešenje zahteva da za svaki broj pronalazite odvojenu formulu, dok bi kompjutersko rešenje podrazumevalo formiranje odgovarajućeg algoritma. Dopustićemo sebi jednu izjavu — postoji algoritam pomoću koga bilo koji prirodan broj može da se prikaže pomoću četiri četvorke, nevolja sa ovim algoritmom je što, koliko nam je poznato, nigde nije publikovan!

Uspeo sam da pomoću četiri četvorke predstavim sve prirodne brojeve manje od ____________.

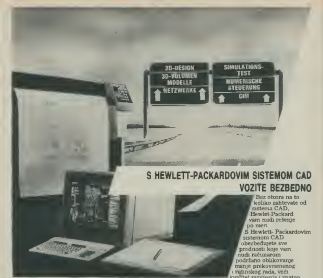
Ime i prezime ____________

Adresa ____________

Mesto ____________

Računar ____________ Vreme ____________

Identifikacioni broj: |_| |_| |_| (iz broja tel.) |_| |_| (god.rođ.) |_| |_| (po izboru)



S HEWLETT-PACKARDOVIM SISTEMOM CAD VOZITE BEZBEDNO

Bez obzira na to koliko zahtevate od sistema CAD, Hewlet-Packard vam nudi rešenje po meri S Hewlett- Packardovim sistemom CAD obezbeđujete sve prednosti koje vam nudi računarom podržano oblikovanje manje prekovremenog i rutinskog rada, veći kvalitet razvijanja i znatno skracena vremena razvoja Odlučujuca razlika u poređenju s drugim sistemima jeste ta, da radite kao konstruktor u ambijentu na koji ste navikli i s jednakim konceptima, pomoćnim sredstvima i oruđima, kao što ste ih dosad upotrebljavali.

Sistemi CAD su integrisani sastavni delovi koncepta HEWLETT-PACKARDOVOG DESIGNCENTRA Time je obezbeđeno da vaš današnji sistem CAD na svakom stepenu usavršanosti i veličine nastavija rad bez prilagođavanja, tako da se može bez problema integrisati u budući koncept CIM.

Potpuno sami odlučujete na kom stepenu tehnike CAD ćete se uključiti u ovaj sistem i u kakvim ćete etapama dopunjavati sistem koji je otvoren nagore Kod Hewlett-Packardovih sistema CAD orijentacija je pravilna, a svoje ciljeve ostvarićete bez skupih krivina

Detaljne informacije možete dobiti pod šifrom »SISTEMI CAD«

Hewlett-Packard G.m.b.H , Lieblgasse 1, 1222 WIEN, tel· 0222/25 00-0
Hermes zastupstvo Hewlett-Packard, 61000 Ljubljana, Celovška 73, Tel 061/552- 941, 559-441, Tlx 31583
Hermes zastupstvo Hewlett-Packard, 11000 Beograd, Generala Ždanova 4, Tel 011/340 327, 342 641, Tlx 11433
Hermes zastupstvo Hewlett-Packard, 71000 Sarajevo, Kralja Tomislava 1, Tel 071-23982/36859, Tlx. 41634

HEWLETT PACKARD

NA GODIŠNJI ODMOR SA BIGZ-ovim KNJIGAMA

BIGZ

Izbor od 100 zanimljivih knjiga po ceni od 70 do 7.500 dinara

Cena kompleta 144 190 dinara Pouzećem 115.352 dinara. Na desetomesečni kredit 18.667 dinara mesečna rata, sa uračunatom kamatom

Br.	Knjiga	Cena
1	Margaret Jurnesar: **OSMEH KRALJEVIĆA MARKA**, pripovetke	70
2	Bulat Okudžava: **GUTLJAJ SLOBODE**, roman	100
3	Bobin Pavlovski: **HOTEL OUVA**, roman	100
4	Kosta Dimitrijević: **NAIVA U JUGOSLAVIJI**, pregled naivaca u Jugoslaviji	100
5	Vladimir Predić: **SEVERNI PROZORI**, pesme	100
6	Maksimilijan Kocijančič: **SPREČITE INFARKT**	180
7	Adam Puslojić: **OARODAVAC**, pesme	170
8	Mona Nade: **ISTORIJA NADREALIZMA**	200
9	Milovan Danojlić: **KAKO JE DOBROSLAV PROTRČAO KROZ JUGOSLAVIJU**, povest	200
10	Pavle Zorić: **DUH ROMANA**, eseji	250
11	Duško Novaković: **NADZORNIK KVARTA**, pesme	300
12	Deni Didro: **RAMOOV SINOVAC O SVOJSTVIMA**	300
13	Ivan Ivanović: **CRVENI KNALJ**, roman	300
14	Predrag Matvejević: **JUGOSLOVENSTVO DANAS**	300
15	Rabindranat Tagore: **GRADINAR**, pesme	300
16	Žak Prever: **NEKE STVARI I OSTALO**, pesme	300
17	**SLOVO O POLKU IGOROVU**, ep	300
18	Dragomir Brajković: **LEDENE GORE, JUŽNA MORA**, pesme	300
19	Goran Babić: **MIŽA ZMIJA**, pesme	300
20	Slobodan Zubanović: **DOMAĆI DUH**, pesme	300
21	Ismet Rebronja: **SREDA I SREDA KĆI**, pesme	300
22	Božidar Šujica: **IZGLEDI ZA SUTRA**, pesme	300
23	Henri Miler: **RAKOVA OBRATNICA**, roman	400
24	Čedomir Marković: **PISCI KNJIGE, ČITAOCI**	400
25	Desimir Blagojević: **IŠČEKIVANJE SJAJA I DOBRODOŠLICA**, pesme	400
26	Jevrem Brković: **KUČKINI SINOVI**, pesme	400
27	Milan Milišić: **MAČKA NA SMEĆU**, pesme	400
28	Ljubica Miletić: **FISON, VODA SENOVITA**, pesme	400
29	Milan Nenadić: **PESME**	400
30	Milorad Durić: **USAMLJENOST**, pesme	400
31	Slobodan Pavićević: **RADOVI NA PUTU**, pesme	400
32	Svetlana Velmar-Janković: **DORĆOL**, pripovetke	400
33	Radomir Smiljanić: **NEKO JE OKLEVETAO HEGELA**, roman	700
34	Milovan Vitezović: **SRCE MI JE OTKUCALO**, aforizmi	700
35	Radmila Lazić: **NOĆNI RAZGOVORI**, pesme	700
36	Miloslav Tešić: **KUPIROVO**, pesme	700
37	Ranko Jovović: **ODMILANJE STRAHA**, pesme	700
38	Rajko Đurić: **HEFESTOVI UČENICI**, pesme	700
39	Branislav Prelević: **ŠESTO ČULO**, pesme	700
40	Miladin Mićeta: **GLUHO**, pripovetke	400
41	Novica Petković: **OD FORMALIZMA DA SEMIOTICI**, ogledi	800
42	Platon: **MENEKSEN-FILEB-KRITIJA**	400
43	Biljana Jovanović: **DUŠA JEDINICA MOJA**, roman	900
44	Aleksandar Nenadović, Dušan Simić: **AMERIKA NA RASKRŠĆU**	1 000
45	Ljubomir Simović: **DNAME** (Hasanagin ca Čudo u Šargenu)	1 000
46	Drago Kekanović: **IVANJSKA NOĆ**, roman	1 100
47	Karlos Kastaneda: **ORLOV DAR**, roman	1 100
48	Milan Oklopdžić: **CA BLUES**, roman	1 100
49	Živojin Pavlović: **ONI VIŠE NE POSTOJE**, roman	1 100
50	Lazar Vučković: **PESME**	1 200
51	**MRSNE PRIČE**, izbor narodne erotske proze	1.300
52	Predrag Matvejević: **RAZGOVORI S KRLEŽOM**	1 300
53	Matija Bećković: **O MEĐUVREMENU**, satirični zapisi	1 300
54	Dragoslav Mihailović: **KAD SU CVETALE TIKVE**, roman	1.300
55	Danilo Kiš: **GROBNICA ZA BORISA DAVIDOVIĆA**, roman	1 300
56	Branimir Šćepanović: **USTA PUNA ZEMLJE**, roman	1 300
57	Džordž Orvel: **ŽIVOTINJSKA FARMA**, roman	1 300
58	Borisav Stanković: **NEČISTA KRV**, roman	1 300
59	Edgar Volas: **OSVETNIK**, kriminalistički roman	1 300
60	A. K. Dojl: **BASKERVILSKI PAS**, kriminalistički roman	1 300
61	**SANOVNIK** sa horoskopima	1 300
62	A. Dima: **DAMA S KAMELIJAMA**, roman	1 500
63	Dušan Radović: **BEOGRADE DOBRO JUTRO I**	1 500
64	Dušan Radović: **BEOGRADE DOBRO JUTRO II**	1 500
65	Dušan Radović: **BEOGRADE DOBRO JUTRO III**	1 500
66	**SRPSKE JUNAČKE PJESME**, zbor od 50 pesama	1 500
67	Jovan Deretić: **KRATKA ISTORIJA SRPSKE KNJIŽEVNOSTI**	1 500
68	Ivo Andrić: **NA DRINI ĆUPRIJA**, roman	1 500
69	Meša Selimović: **DERVIŠ I SMRT**, roman	1 500
70	Erika Jong: **STRAH OD LETENJA**, roman	1 500
71	Živojin Pavlović: **ZADAH TELA**, roman	1 500
72	**BRZI KUVAR**	1 300
73	Oktavio Paz: **HIMNA MEĐU RUŠEVINAMA**, pesme	1 400
74	Volt Vitmen: **VLATI TRAVE**, pesme	1 400
75	I. B. Singer: **ROB**, roman	1 700
76	Miodrag Bulatović: **LJUDI SA ČETIRI PRSTA**, roman	1 700
77	Milisav Savić: **TOPOLA NA TERASI**, roman	1 750
78	Moma Marković: **SAZREVANJE REVOLUCIJE** (Sećanja 1936—1941)	1.800
79	**SAVREMENA POLJSKA POEZIJA**	2 000
80	Borislav Pekić: **BESNILO**, roman	2 000
81	Platon: **ION-FEDAR-GOZBA**	2 000
82	Platon: **ODBRANA SOKRATOVA — KRITON — FEDON**	2 000
83	Milan Komnenić: **IZGON**, pesme	2 100
84	Hana Dalipi: **VIKEND U MATERINI**, roman	2.500
85	Dragoljub Pavlović: **PITANJEM NA ODGOVORE**	2.500
86	Sveta Lukić: **RATNE IGRE U VRBOVCU**, roman	2 400
87	B. Mihajlović Mihiz: **IZDAJICE**, drama	2 400
88	Aleksandar Popović: **MREŠĆENJE ŠARANA** i druge drame	2 400
89	Hatidža Krnjević: **LIRSKI ISTOČNICI** (Iz istorije i poetike lirske narodne poezije)	3 000
90	Đuro Damjanović: **ZEMUNSKA HRONIKA**, roman	3 200
91	Klod Levi-Stros: **MITOLOGIKE I—III**	3 200
92	Dobrica Ćosić: **GREŠNIK**, roman	3 500
93	Stevan Raičković: **SABRANA POEZIJA** u 6 knjiga	3 500
94	Petar Džadžić: **HOMO BALCANICUS HOMO HEROICUS**, eseji	5 000
95	Rafael Alberti: **PESME**	5.000
96	Fransoa Vijon: **ZAVEŠTANJA**, pesme	5 000
97	Rajko Đurić: **SEOBE ROMA**	5 000
98	Dobrica Ćosić: **OTPADNIK**, roman	5 600
99	Mirko Kovač: **REBESKI ZARUČNICI**, pripovetke	6 000
100	Vladislava Ribnikar: **MOGUĆNOSTI PRIPOVEDAČA**, ogled o novijoj srpskoj prozi (Antonije Isaković, Dragoslav Mihailović, Mirko Kovač, Vidosav Stevanović, Milisav Savić, Miroslav Josić Višnjić, Radoslav Bratić)	7.500

Neporučena preporuka 787
Beogradski izdavačko-grafički zavod
11001 Beograd, Bulevar vojvode Mišića 17
poštanski fah 348, telefon 653-768

(Mesto i datum)

Poručujem
— **KOMPLET OD 100 BIGZ-ovih knjiga** cena 144 190 dinara
— **pojedine knjige** iz kompleta

(navesti redne brojeve knjiga)

Ukupan iznos porudžbine od ______ dinara platiti:
a) **odjednom** pouzećem (plaćanje poštaru prilikom prijema knjiga), sa 20% popusta ako je vrednost porudžbine više od 4 000 dinara
b) u ______ **mesečnih rata** (najviše 10 rata, najmanji iznos rate 1.500 dinara), sa odgovarajućom kamatom, pri čemu ću prvu ratu platiti poštaru prilikom prijema knjiga, a ostale rate uplatnicama koje ću dobiti od BIGZ-a

Knjige se isporučuju odmah.
U slučaju spora nadležan je odgovarajući sud u Beogradu

(Prezime, ime oca i ime)

(Zanimanje) (Telefon u stanu — telefon na poslu)

(Adresa stana: broj pošte, mesto, ulica i broj)

(Organizacija u kojoj je poručilac zaposlen i njena adresa)

Overa o zaposlenju, penzioneri prilažu pretposlednji ček penzije

Potpis poručioca, broj lične karte i mesto izdavanja

KOD PORUDŽBINA MANJIH OD 2 000 DINARA ZARAČUNAVA SE 500 DINARA ZA POŠTARINU I EKSPEDICIJU

NEOVERENE PORUDŽBINE (pri plaćanju u ratama) **NE PRIMAMO!**

Prevođenje brojeve

Brojevi iz šešira

U „Računarima" smo do sada često predstavljali algoritme velikih majstora programiranja. U svakodnevnom radu, međutim, programeri se najčešće sreću sa trivijalnim problemima koji se rešavaju trivijalnim algoritmima, ali put do njih, i pored sve jednostavnosti, ponekad nije nimalo lak. Imajući u vidu upravo potrebe svakodnevnog programiranja, Jovan Skuljan je pripremio seriju tekstova „Svakodnevni algoritmi", koji će se na kraju sklopiti u univerzalni interpreter programskog jezika.

Prevođenje brojeva iz jednog brojnog sistema u drugi je problem koji se javlja prilično često u toku pisanja programa. Najčešće imamo potrebu da prevodimo binarne i heksadekadne brojeve u decimalne i obrnuto, iz jednostavnog razloga što računar ne koristi brojni sistem sa osnovom deset a čoveku, s druge strane, nisu naročito bliski sistemi sa osnovom dva ili šesnaest.

Cilj nam je da u ovom tekstu kažemo nešto više o sistemima brojanja uopšte, kao i o načinu prevođenja brojeva iz jednog sistema u drugi.

Brojem označavamo količinu nečega. Možemo brojati kruške, ovce, dane, korake, pare, ili bilo šta drugo. Čovek se susreo sa problemom brojanja sasvim sigurno pre nego što je naučio da zapisuje brojeve.

Danas već i deca predškolskog uzrasta umeju da broje barem do deset. Taj magični broj deset tako od prvih godina života počinje da se uvlači u našu svest i podsvest, pa kasnije gotovo da i ne primećujemo koliko smo vezani za njega kad obavljamo bilo koju operaciju sa brojevima. A o čemu se zapravo radi?

Kako brojimo?

Iz davne prošlosti smo nasledili „običaj" da sve brojeve zapisujemo koristeći samo *deset osnovnih znakova:* 0, 1, 2, 3, 4, 5, 6, 7, 8 i 9. Ti znakovi, ili *cifre*, označavaju, svaka za sebe odgovarajuću *količinu*. Na primer, zamislimo da brojimo neke *kruške* sa gomile i bacamo ih u korpu. Postupak je sledeći:

Prevođenje broja u binarni zapis

1. Potprogram NUMBIN uzima osmobitni broj sa adrese HL, čita redom njegove bitove i formira odgovarajuće cifre binarnog zapisa unazad počev od adrese DE. Na izlazu HL sadrži adresu sledećeg bajta, a DE adresu sledeće cifre.

```
NUMBIN  LD    C (HL)       Uzima se broj sa adrese HL.
        LD    B #08        Priprema se brojač osam bitova.
NBITS   XOR   A            Na početku petlje akumulator se
                           postavlja na nulu.
        RR    C            Pomeranjem udesno bitovi se redom
                           dovode u indikator prenosa. Prvo
                           bitovi manje težine
        RLA                Indikator prenosa dovodi se u
                           akumulator. Sada je akumulator
                           jedan ili nula, zavisno od toga da li
                           je indikator prenosa setovan ili ne.
        ADD   A #30        Formiranje ASCII koda (nula se
                           kodira sa #30)
        LD    (DE) A       Cifra se smešta u memoriju.
        DEC   DE           Priprema pozicije za sledeću cifru
        DJNZ  NBITS        Ponavljanje za osam bitova.
        INC   HL           Priprema adresa sledećeg bajta
        RET                Povratak
```

2. Program NBIN8 prevodi osmobitni broj sa adrese NUMBER u binarni niz cifara počev od adrese STRING

```
NBIN8   LD    HL NUMBER    Uzima se adresa broja
        LD    DE STRING+7  Adresa poslednje cifre
        CALL  NUMBIN       Vrši se prevođenje broja
        RET                Povratak.
```

3. Program NBIN16 prevodi 16-bitni broj sa adrese NUMBER u binarni zapis. Niz od 16 ASCII cifara smešta se u memoriju počev od adrese STRING

```
NBIN16  LD    HL NUMBER    Uzima se adresa broja.
        LD    DE STRING+15 Adresa poslednje cifre
        CALL  NUMBIN       Prevodi se bajt manje težine
        CALL  NUMBIN       Prevodi se bajt veće težine
        RET                Povratak.
```

Prevođenje broja u heksadekadni zapis

1. Potprogram DIGHEX formira ASCII znak heksadekadne cifre sadržane u niža četiri bita registra C. Tako dobijeni bajt smešta se na adresu DE. Na izlazu DE se pomera unazad na sledeću cifru. Registar C ostaje očuvan.

```
DIGHEX  LD    A C          Uzima se zadati broj
        AND   #0F          Izdvajaju se niža četiri bita
        ADD   A,#30        Formiranje ASCII koda. Dodavanje
                           #30 dovoljno je samo za cifre 0 —
                           9
        CP    #3A          Da li je cifra veća od „9"?
        JR    C CODFIN     Ako nije, vrši se skok napred
        ADD   A #07        Formira se odgovarajući kôd za
                           slova A — F
CODFIN  LD    (DE) A       Cifra se smešta u memoriju
        DEC   DE           Priprema se pozicija za sledeću
                           cifru.
        RET                Povratak.
```

2. Potprogram NUMHEX uzima osmobitni broj sa adrese HL i uzastopnim pozivom DIGHEX formira odgovarajuće dve heksadekadne cifre. Na ulazu HL pokazuje broj u memoriji a DE poziciju cifre manje težine. Na izlazu HL pokazuje sledeći bajt, a DE sledeću cifru unazad

```
NUMHEX  LD    C (HL)       Uzima se bajt sa zadate adrese.
        INC   HL           Priprema sledeće adresa
        CALL  DIGHEX       Formira se cifra manje težine
        RRC   C            Uzastopnom rotacijom
        RRC   C            registra C četiri puta udesno
        RRC   C            vrši se inverzija
        RRC   C            heksadekadnih cifara
        CALL  DIGHEX       Formira se cifra veće težine
        RET                Povratak.
```

3. Program NHEX8 prevodi osmobitni broj sa adrese NUMBER u heksadekadni zapis na adresi STRING

```
NHEX8   LD    HL NUMBER    Uzima se adresa broja.
        LD    DE STRING+1  Adresa poslednje cifre
        CALL  NUMHEX       Prevođenje broja
        RET                Povratak.
```

4. Program NHEX16 prevodi 16-bitni broj sa adrese NUMBER u heksadekadni niz od četiri cifre na adresi STRING

```
NHEX16  LD    HL NUMBER    Uzima se adresa broja
        LD    DE STRING+1  Adresa poslednje cifre
        CALL  NUMHEX       Prevodi se bajt manje težine
        CALL  NUMHEX       Prevodi se bajt veće težine
        RET                Povratak.
```

Dok je korpa prazna, kažemo „nema ništa" i pišemo „0". Čim ubacimo jednu krušku kažemo „jedan" i pišemo „1". Zatim ubacimo još jednu krušku i kažemo „dva" a pišemo „2". Tako redom ubacujemo i brojimo kruške dok ih ne bude devet, kada ćemo pisati „9".

Time smo iscrpli svih deset cifara. Kruške i dalje možemo ubacivati u korpu i brojati, ali zapis neće biti tako jednostavan. Odgovor je sledeći: pišemo cifru „1" kao znak da smo već jednom iskoristili postojeći skup cifara brojeći od nule do devet, a brojanje nastavljamo kao i ranije. Biće dakle, 10 krušaka pa 11 krušaka itd. do 19.

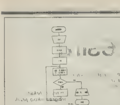
Onda pišemo „2", a brojimo opet od 0 do 9: 20, 21, ... 29.

Brojanjem do 99 iskoristićemo deset puta postojeći skup cifara. Zato dalje pišemo „10" i brojimo ponovo od 0 do 9: 100, 101, 102, ... 109.

I tako — u beskonačnost.

Kako bismo mogli da brojimo?

Sve što smo do sada rekli, i te kako je dobro poznato i o tome nema šta mnogo da se priče. Ono što je za nas sada interesantnije



Slika 1 Algoritam konverzije broja

jeste da brojanje, samo po sebi nije problem *ukoliko nam je poznat skup cifara* i pravilo po koma se cifre rađaju

Odmah se nameće interesantno pitanje Zašto baš deset cifare? Da li ih može biti više od deset? Da li ih može biti manje od deset?

Nećemo se ovde upuštati u rasprave oko toga zašto čovek koristi baš deset cifara Ali zato odmah možemo dati odgovor na druga dva pitanja sasvim je svejedno barem kada se posmatre sa farmalne tačke gledišta, koliko cifare imamo na raspolaganju Minimum su dve jedna koja označava „ništa i druga koja označava „jedan Ali, može ih biti i stotinu U tom slučaju bismo svaku količinu od nule do devedesetdevet označavali jednom jedinom cifrom Tek broj „sto" bio bi zapisan kombinacijom dve cifre

Uzećemo kao primer skup od osam cifara 0 1, 2 3 4 5, 6 i 7 Brojanje od nule do sedam u takvom sistemu ni po čemu se ne razlikuje od brojanje sa deset cifara Međutim broj osam bismo u tom slučaju morali pisati kao „10" jer smo jednom već iskoristili čitav skup cifara i otpočeli ponovo brojanje od nule Dalje bismo brojali devet" a pisali 11" brojali deset" a pisali „12" itd

Šta bi recimo u takvom sistemu označavao broj 47 ? U sistemu sa deset cifara rekli bismo da je to četrdeset sedam" jer smo četiri puta izbrojali po deset i još stigli do sedam Ali kada radimo sa osam cifara onda to znači da smo **četiri puta brojali po osam** (što je tridesetdva) plus sedam što je ukupno **tridesettiavet.**

Zvuči možda komplikovano i pomalo neobično da broj 47 čitamo kao tridesetdevat" ali sve je to stvar navike

Prevođenje broja u decimalni zapis

1 Potprogram DIV formira cifre decimalnog broja uzastopnim deljenjem sa 10^k To nije uobičajen algoritam ali je podesan kada se zna koliko cifara broj može najviše imati Ograničili smo se samo na 16-bitne brojeve a to znači da u decimalnom zapisu nećemo imati više od pet cifara

Na ulazu HL sadrži zadati broj ili ostatak iz prethodnog deljenja a BC ima vrednost -10^k Deljenje se vrši uzastopnim oduzimanjem odnosno sabiranjem pošto je BC negativno Rezultujuća decimalna cifra se smešta na adresu DE Prvo se računaju cifre veće težine Na izlazu HL sadrži ostatak deljenja a DE pokazuje poziciju sledeće cifre

```
DIV     LD    A #30        Priprema akumulatora za ASCII
                           kodiranje
REPT    ADD   HL,BC        Petlja za uzastopno oduzimanje
        INC   A            Kodovi se formiraju brojanjem
                           ciklusa
        JR    C REPT       Postupak se ponavlja sve dok se BC
                           sadrži u HL
        SBC   HL BC        Vrši se korekcija sadržaja HL i
        DEC   A            akumulatora zbog poslednjeg
                           oduzimanja
        LD    (DE) A       Cifra se smešta u memoriju
        INC   DE           Priprema se adresa sledeće cifre
        RET                Povratak.
```

2 Program NDEC16 prevodi 16-bitni broj sa adrese NUMBER u niz od pet decimalnih cifara na adresi STRING sa vodećim nulama

```
NDEC16  LD    HL (NUMBER)  Uzima se zadati broj
        LD    DE,STRING    Uzima se adresa niza cifara
        LD    BC -10000    Priprema registra BC i
        CALL  DIV          formiranje prve cifre
        LD    BC -1000     Priprema registra BC i
        CALL  DIV          formiranje druge cifre
        LD    BC -100      Priprema registra BC i
        CALL  DIV          formiranje treće cifre
        LD    BC -10       Priprema registra BC i
        CALL  DIV          formiranje četvrte cifre
        LD    A L          Preostala peta cifra formira se
                           prostim
        ADD   A #30        prevođenjem u ASCII zapis.
        LD    DE,A         Peta cifra se smešta u memoriju
        RET                Povratak.
```

Prevođenje broja iz binarnog zapisa

1 Potprogram BINNUM čita binarne ASCII cifre počev od adrese DE i na osnovu toga formira vrednost osmobitnog broja na adresi HL Na ulazu DE pokazuje cifru najmanje težine a HL adresu broja Na izlazu registar DE je spreman za sledeću cifru veće težine a HL za sledeći bajt.

```
BINNUM  LD    B #08        Priprema brojača osam bitova
BBITS   LD    A (DE)       U ciklusu se uzimaju cifre iz
                           memorije
        CP    #31          Da li je cifra 0' ili 1 ?
        CCF                Indikator prenosa biće sada setovan
                           za 1 a resetovan za 0
        RR    C            Dobijeni bit se unosi u registar C
        DEC   DE           Prelaz na sledeću cifru
        DJNZ  BBITS        Postupak se ponavlja osam puta
        LD    (HL) C       Dobijeni broj se unosi u memoriju
        INC   HL           Prelaz na sledeći bajt.
        RET                Povratak.
```

2 Program BNUM8 formira osmobitni broj na adresi NUMBER na osnovu binarnog zapisa sa adrese STRING

```
BNUM8   LD    HL NUMBER    Uzima se adresa broja
        LD    DE STRING+7  Uzima se adresa poslednje cifre
        CALL  BINNUM       Prevođenje niza cifara
        RET                Povratak
```

3 Program BNUM16 formira 16-bitni broj na adresi NUMBER na osnovu binarnog zapisa sa adrese STRING

```
BNUM16  LD    HL NUMBER     Uzima se adresa broja
        LD    DE STRING+15  Uzima se adresa poslednje cifre
        CALL  BINNUM        Prevodi se niži bajt broja
        CALL  BINNUM        Prevodi se viši bajt broja
        RET                 Povratak
```

Osnova brojnog sistema

Broj cifara koje koristimo pri zapisivanju brojeva predstavlja *osnovu brojnog sistema* Tako kažemo da čovek koristi brojni sistem sa osnovom deset, a videli smo i kako bi izgledalo brojanje u sistemu sa osnovom osam Ne postoji nikakva principijelna razlika ni u sistemu sa bilo kojom drugom osnovom

Uvešćemo zato sada neki proizvoljan brojni sistem sa osnovom n Šta je to n ? To je bilo koji po volji izabran prirodan broj veći od jedan Možemo slobodno uzeti umesto „n recimo konkretan broj 16 i onda svuda gde vidimo n pišemo 16 ali smatramo da elementarna matematička notacija našim čitaocima nije toliko strana

Naš sistem brojanja ima skup od tačno n cifara koje ćemo označavati sa x Svaki broj je zapisan kao neki niz cifara Na primer

$$x_3x_2x_1x_0$$

indeksi 0 1 2 3 označavaju poziciju cifre u zapisu broja Jasno je da cifra x_3 i x_0 na primer ne doprinose podjednako vrednosti broja već se razlikuju po težini Cifra x_3 nosi veću težinu Uzmimo konkretan primer broja *2417* U sistemu sa osnovom deset cifra 2" označava ustvari „dva hiljade i ima veću težinu od cifre „7" koja označava jedinice

Težina cifarskog mesta iznosi n^k gde je k udaljenost cifre od desnog kraja Na primer cifre x_3 ima težinu n^3 Vrednost broja dobije se prostim zbirom

$$A = x_3n^3 + x_2n^2 + x_1n^1 + x_0n^0$$

Dogovorićemo se da bez obzira na vrednost osnove n cifre brojnog sistema zapisujemo poznatim znakovima 0 1 2, 3 itd Ako je osnova veća od deset tako da su nam potrebne nove cifre preko 9 koristićemo slova abecede A B C, D Ovim smo se obezbedili za sva brojna sistema čija osnova ne prelazi četrdeset (!) Sistemi sa većom osnovom nam sasvim sigurno nikad neće biti potrebni

Interesantne osobina

Videli smo da se broj „deset u sistemu sa osnovom deset zapisuje kao 10" Isto tako se broj „osam u sistemu sa osnovom osam zapisuje se 10 Nije teško videti da će se bilo koji broj n u sistemu sa osnovom n takođe zapisivati kao „10" jer smo rekli da po dogovoru taj zapis znači da smo jednom već iskoristili sve n cifara i počeli da brojimo od nule

Matematička operacija nad brojevima postaju izuzetno jednostavne ako se radi sa osnovom brojnog sistema. Zašto je nama lako da množimo i delimo sa deset? Naravno, zato što je deset osnova našeg sistema.

Množenje osnovom sistema vrši se **prostim dodavanjem nule** broju koji množimo. Tako je, na primer, *47×10=470*. Međutim, interesantno je da ovaj identitet važi u istom obliku na samo u sistemu sa osnovom deset (!). Recimo, u sistemu sa osnovom osam, to bismo pročitali kao „tridesetdevet puta osam jeste tristadvanaest".

Razlomljeni decimalni brojevi pišu se pomoću *decimalnog zareza* ili tačke. Zapis *47.3* na primer u sistemu sa osnovom deset označava broj „četrdesetsedam celih i tri desete dela od celog". U sistemu sa osnovom osam to bi bilo „tridesetdevet celih i tri osme dela od celog". Množenje osnovom sistema samo pomera decimalni zarez udesno za jedno mesto. Deljenje osnovom sistema pomera zarez ulevo.

Iz sistema u sistem

Problem koji nas ovde najviše interesuje je kako prevesti neki broj iz zapisa u sistemu sa osnovom *m* u odgovarajući zapis u sistemu sa osnovom *n*. Na primer, kako prevesti broj „A7DE" iz heksadekadnog sistema (sistem sa šesnaest cifara: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E i F) u decimalni sistem? Ili, kako broj „42974" prevesti iz decimalnog u heksadekadni sistem?

Neka je dat ceo broj „A" u sistemu sa osnovom *m*. Zapis tog broja zadat je nizom cifara $Y_{L-1}Y_L \ldots Y_2Y_1Y_0$. Nas interesuje zapis tog broja u sistemu sa osnovom *n*, a to će biti neki drugi niz cifara: $x_{k-1}x_k \ldots x_2x_1x_0$. Tražimo, u stvari, algoritam za nalaženje cifara *x*.

Postupak je univerzalan i sastoji se u sledećem:

1. Delimo broj *A* osnovom sistema *n* u koji broj prevodimo. Naravno, operaciju deljenja vršimo *u sistemu sa osnovom m*, jer je i broj zapisan u tom sistemu.

2. Označimo ceo deo količnika sa *C*. Ostatak koji dobijamo pri deljenju brojno je jednak samoj cifri x_0 u zapisu koji tražimo.

Pravođenje broja iz heksadekadnog zapisa

1. Potprogram HEXDIG formira u akumulatoru vrednost koja odgovara heksadekadnoj cifri sa adrese DE. Na izlazu DE pokazuje na sledeću cifru veće težine.

```
HEXDIG  LD    A,(DE)       Uzima se ASCII cifra iz memorije
        DEC   DE           Pomeranje na sledeću cifru.
        SUB   #30          Formira se vrednost cifre
                           Oduzimanje #30 je dovoljno samo
                           za cifre 0 — 9
        CP    #0A          Da li je cifra veća od „9"?
        RET   C            Ako nije, povratak
        SUB   #07          Korigovanje koda za slova abecede
                           A — F
        RET                Povratak.
```

2. Potprogram HEXNUM uzastopnim pozivom HEXDIG formira na adresi HL vrednost osmobitnog broja zapisanog nizom cifara na adresi DE. Na ulazu HL pokazuje poziciju broja a DE cifru najmanje težine. Na izlazu se HL pomera na sledeći bajt a DE unazad na sledeću cifru.

```
HEXNUM  CALL  HEXDIG       Formira se cifra manje težine
        LD    C,A          Rezultat se čuva u registru C
        CALL  HEXDIG       Formira se cifra veće težine
        RLCA               Uzastopnom cikličnom rotacijom
                           akumulatora
        RLCA               četiri puta ulevo
        RLCA               cifra veće težine se
        RLCA               dovodi na svoje pravo mesto
        OR    C            Uzima se i cifra manje težine
        LD    (HL),A       Broj se smešta u memoriju.
        INC   HL           Pozicija sledećeg bajta.
        RET                Povratak.
```

3. Program HNUM8 formira osmobitni broj na adresi NUMBER na osnovu heksadekadnog zapisa na adresi STRING.

```
HNUM8   LD    HL,NUMBER    Uzima se adresa broja
        LD    DE,STRING+1  Uzima se adresa poslednje cifre
        CALL  HEXNUM       Prevodi se zadati niz cifara
        RET                Povratak.
```

4. Program HNUM16 formira 16-bitni broj na adresi NUMBER, na osnovu heksadekadnog zapisa na adresi STRING.

```
HNUM16  LD    HL,NUMBER    Uzima se adresa broja
        LD    DE,STRING+3  Uzima se adresa poslednje cifre
        CALL  HEXNUM       Prevodi se bajt manje težine
        CALL  HEXNUM       Prevodi se bajt veće težine.
        RET                Povratak.
```

Pravođenje broja iz decimalnog zapisa

1. Potprogram MULT obavlja operaciju HL=10*HL+A, čime se postojećoj vrednosti HL dodaje nova dekadna cifra A.

```
MULT    LD    C,L          Sadržaj registra HL se
        LD    B,H          prenosi i u BC
        ADD   HL,HL        Množenje sa dva
        ADD   HL,HL        Množenje sa četiri
        ADD   HL,BC        Množenje sa pet.
        ADD   HL,HL        Množenje sa deset
        LD    B,#00        Sadržaj akumulatora se
        LD    C,A          prenosi u BC
        ADD   HL,BC        Završna operacija sabiranja.
        RET                Povratak.
```

2. Potprogram DECIM uzastopnim pozivom MULT formira u HL vrednost 16-bitnog broja čiji decimalni zapis počinje na adresi DE. Zapis može imati između jedne i pet cifara, iza kojih obavezno mora postojati graničnik u vidu proizvoljnog bajta sa kodom manjim od #30.

```
DECIM   LD    HL,#0000     Priprema HL za ulaz u petlju.
DECDIG  LD    A,(DE)       Redom se uzimaju cifre iz ASCII niza
                           i
        SUB   #30          formira se njihova numerička
                           vrednost.
        RET   C            Povratak ako je dostignut graničnik.
        CALL  MULT         Nova cifra se dodaje broju.
        INC   DE           Prelaz na sledeću cifru.
        JR    DECDIG       Nazad u petlju
```

3. Program DECNUM formira 16-bitni broj na adresi NUMBER na osnovu decimalnog zapisa sa adrese STRING. Zapis ima oblik ASCII niza sa graničnim bajtom.

```
DECNUM  LD    DE,STRING    Uzima se adresa niza cifara
        CALL  DECIM        Formira se vrednost broja
        LD    (NUMBER),HL  Broj se smešta u memoriju
        RET                Povratak.
```

3. Delimo dalje *C* sa *n*. Ostatak koji sada dobijamo biće brojno jednak x_1, a novi ceo deo količnika ponovo delimo sa *n*, u ostatku dobijamo x_2 itd.

Postupak se ponavlja sve dok ceo deo količnika ne postane nula. Algoritam je dat na slici 1. Neka posebna objašnjenja nisu potrebna, ali moramo se zadržati na jednom suštinskom problemu. Kada kažemo da je na ulazu zadat broj *A*, onda, u stvari, podrazumevamo da je zadat *niz cifara y* kao zapis tog broja *A* u sistemu sa osnovom *m*. Operacije celobrojnog deljenja sa ostatkom *C DIV n* i *C MOD n* moraju se izvoditi u sistemu sa osnovom m. U tom slučaju, neophodno je razraditi odgovarajući algoritam, što donekle usložnjava jednostavnost postupka konverzije.

Postoji i drugi metod konverzije koji zahteva izvođenje operacija u sistemu u koji se broj prevodi a ne u onom iz kojega se broj prevodi. Prosto, svaku cifru broja *A* treba množiti odgovarajućom težinom i dobijene rezultate sabrati:

$$A = Y_{L-1}n^{L-1} + Y_{L-2}n^{L-2} + \ldots + Y_2n^2 + Y_1n^1 + Y_0n^0$$

Na ovaj način ne dobijamo u rezultatu niz cifara *x*, već dobijamo broj *A* „preračunavajući njegovu vrednost" u sistemu sa osnovom *n*. Ovo je najpodesniji metod za prevođenje brojeva u decimalni sistem, jer tada vršimo operacije isključivo u decimalnom sistemu, što umemo da radimo bez problema. Obrnutu konverziju iz decimalnog sistema u neki drugi vršimo po algoritmu sa slike 1, jer opet izvodimo operacije u sistemu sa osnovom deset.

Računar će raditi slično, ali sve u sistemu sa osnovom dva.

Jednostavni programi

Kao ilustraciju probleme konverzije brojeva, izabrali smo sledeći konkretan slučaj: dat je neki osmobitni ili šesnaestobitni broj, smešten u memoriji računara na adresi NUMBER. Odrediti niz ASCII cifara kojima se taj broj zapisuje u sistemima sa osnovom dva, šesnaest, ili deset i smestiti taj niz u memoriju počev od adrese STRING. I obrnuto, ako je dat niz cifara na adresi STRING, sračunati vrednost broja i smestiti ga na adresu NUMBER.

Primeri su rađeni za mikroprocesor Z80, ali nije veliki problem prevesti ih na neki drugi jezik istog nivoa.

Jovan Skuljan

1. Samo da izvede IRET instrukciju uz eventualno postavljanje neke programske zastavice kako bi se kasnije u toku programa utvrdilo da je ovaj interapt generisan
2. Može da postavi ili resetuje CARRY zastavicu i vrati se u DOS FAR RET naredbom pri čemu DOS ako je Cy=1 prekida program ili za Cy=0 ne preduzima ništa
3. Može da izvrši skok na neku predviđenu labelu u programu na kojoj inače počinje deo za obradu ostalih grešaka prenoseći neki pogodan kod greške Pri tome se nikad ne izvodi povratak iz interapta u DOS Iako ovaj metod deluje pomalo grubo nema nikakve posledice po sistem

Prvi od tri izložena načina je svakako najjednostavniji Preuzimanje interapt rutina 23H nije na žalost dovoljno da se u potpunosti spreči prekid programa usled pritiska na neku od kombinacija stetera. CTRL+BREAK će i dalje obavljati funkciju jer BIOS generiše interapt 1BH i njegova vrednost mora biti izmenjena i usmerena na istu novu rutinu

Posebnu pažnju treba obratiti na činjenicu da se vektoru 23H automatski vraće vrednost po završetku rada programa ali da to nije slučaj sa 1BH koma se originalna vrednost mora vratiti pre kraja rada programa

Tipična sekvenca za sprečavanje ovih prekida data je u sledećem primeru

```
CODE        SEGMENT
            MOV     AX 351BH
            INT     21H
            MOV     ORIG1BS ES
            MOV     ORIG1BO BX
            PUSH    DS
            MOV     DX SEG DUMMYC
            MOV     DS DX
            MOV     DX,OFFSET DUMMYC
            MOV     AX 2523H
            INT     21H
            MOV     AX 251BH
            INT     23


DUMMYC      IRET
```

Interapt kritične greške

Ako se zadovoljavate izborom mogućnosti koja inače postoje pri obradi ove greške u COMMAND COM modulu (Abort Retry Ignore?) onda je postupak podjednako jednostavan kao i prethodni

Pri ulasku u rutinu za obradu greške sve je podešeno da DOS nakon što izvedete IRET može na osnovu sadržaja AL registra da odluči koju od tri operacija da izvede

AL=0 — ignoriše grešku (što retko može da zadovolji)
AL=1 — ponovo pokušava da izvede operaciju (što najčešće ponovo daje isti efekat)
AL=2 — prekida program preko interapta 23H (najgore moguće)

Svakako bi najefikasniji metod bio sličan rednom broju 3 kod CTRL-C interapta Međutim dokumentovanost ovakvog metoda je veoma slabe Na steku se nalazi sadržaj svih registara koji su sačuvani pri pozivu funkcija sa INT 21H u kojoj je došlo do greške Na dnu je povratna adresa i status reč od poziva INT 24H a na vrhu iste struktura od poziva INT 21H lapeda da bi skidanjem sa steka 24 reči došli do situacija kakva je postojala odmah nakon poziva DOS funkcija Tada bi IRET instrukcija izvršila povratak u program odmah nakon instrukcija INT 21H pa bi nam samo ostalo da nekom globalnom zastavicom javimo programu šta se u stvari desilo Međutim ovaj metod je u uputstvu za DOS obeležen upozorenjem da takvim tretiranjem DOS ostaje u NESTABILNOM (!?) stanju sve dok se ne izvede neka funkcija broja većeg od 12

* * *

Iako ima ambiciozan naslov „Sve MS DOS funkcije" (malo pesnička sloboda nikada nije na odmet) ovaj dodatak ne donosi pregled *baš svih* MS DOS funkcija Njegova namena međutim i nije u tome da bude iscrpan nego da uvede programere u svet MS DOS-a i mašinskog programiranja na računarima PC

Zoran Životić

Sve MS DOS funkcije

Zahvaljujući računarima kojima je namenjen MICROSOFT DISK OPERATING SISTEM — MS-DOS — je jedan od najrasprostranjenijih operativnih sistema današnjice Interesantno je da autor prve verzije nije firma čije ime nosi — nastao je sredinom 1980 u firmi Seattle Computer Products pod nazivom 86-DOS i predstavljao je CP/M operativni sistem prilagođen novoj šesnaestobitnoj seriji Intelovih mikroprocesora Sličnost sa CP/M-om je bila tolika da je bilo dovoljno program propustiti kroz specijalan prevodilac nakon čega se mogao bez problema koristiti na novim mašinama U potrazi za operativnim sistemom za PC računare koja je upravo tada razvijao IBM se obratio i Majkrosoftu koji u to vreme nije imao ništa da ponudi Zalogaj je očigledno bio velik pa se Majkrosoft potrudio da otkupi 86-DOS i uz neka poboljšanja ponudi IBM-u MS-DOS 1 0 čime je započela apsolutna dominacija ovog u to vreme već pomalo zastarelog operativnog sistema

Sledeće verzije sa znatnim poboljšanjima pojavljivale su se brzo Najveći skok je učinjen 1983 sa verzijom 2 0 koja je omogućavala rad sa tvrdim diskovima do 10 MB i adekvatno tome hijerarhijskom organizacijom direktorija Prošle godine je izbačena poslednja verzija (3 20) koja je i danas u upotrebi ali sa osim podrška nekim novim tipovima disk jedinica i naznakama daljeg razvoja u pravcu multitasking operativnih sistema malo šta bitno promenila u odnosu na verziju 2 0

Uopšteno bi se moglo reći da je MS-DOS prevalio deo puta od CP/M-a ka UNIX-u koma je danas u mnogim tačkama bliži nego svom prvobitnom uzoru Ovakav razvojni put, uz potrebu da se stalno održava kompatibilnost sa ranijim verzijama, nije mogao da prođe bez posledica To se naročito oseća u radu sa datotekama za koje u mnogim tačkama postoji dupliran set funkcija — jedne su ostatak starijih verzija i rade sa svim pa i najnovijim varijantama MS-DOS-a, druge na višem nivou i sa drugačijim pristupom mogu da se koriste samo sa verzijama DOS-a od 2 0 naviše Ovakva situacija ne bi bila ni malo loša kada bismo u potpunosti mogli da zanemarimo bilo koju od ova dva grupe Na žalost u praksi smo ponekad primorani da koristimo funkcije iz obe Iako to zahteva različite pristupe u načinu programiranja Bez obzira na ovu dvojnost DOS se međutim ipak može oceniti kao pouzdan operativni sistem koji se jednostavno i lako koristi

Programerske alatke

Da bismo napisali bilo koji mašinki program pa i najjednostavniji neophodno je da koristimo nekoliko programa Zbog ograničenog prostora dajemo samo osnovna uputstva za rad sa ovim programima

IBM Macro Assembler — MASM

Iako postoji više verzija asemblera (Digital Research RASM Mark Williams AS itd) zbog usaglašenosti sa notacijom koja se koristi u svim drugim IBM ili Majkrosoftovim kompajlerima koji imaju mogućnost korišćenja asemblerskih modula najbolje je raditi sa ovim asemblerom Radi se inače o veoma dobrom programu koji poseduje veliki broj korisnih osobina — od načina preglednijeg i bržeg pisanja preko kreiranja različitih struktura podataka i podrške 8087 matematičkog koprocesora do kreiranja makro naredbi Vreme prevođenja je sa prvim verzijama MASM 1 0) bilo dosta dugačko ali najnovija verzija (MASM 4 0) ispravlja u potpunosti ovaj nedostatak

Kreiranja programa se može obaviti u bilo kom tekst editoru koji tekstove snima na disk kao standardne sekvencijalne datoteke. Uobičajeno je da se za nastavak naziva upotrebljava ASM. Tipična procedura prevođenja teče na sledeći način

```
C>MASM
Microsoft (R) Macro Assembler  Version 4.00
Copyright (C) Microsoft Corp 1981, 1983, 1984,1985. All rights reserved

Source filename [ ASM]: DEMO           | naziv izvornog programa
Object filename [DEMO.OBJ]:            | izlazna datoteka sa objektnim kodom
Source listing  [NUL.LST]:             | formatiran listing izvornog programa
Cross-reference [NUL.CRF]:             | datoteka za CREF program

  48612 Bytes symbol space free

      0 Warning Errors
      3 Severe  Errors

C>
```

Slika 1

U zagradama MASM daje inicijalne vrednosti koje će biti prihvaćene ako se na pitanja odgovori samo sa RETURN. NUL datoteka (bez obzira na produžetak naziva) je simulacija nepostojećeg izlaznog uređaja i ne upisuje se na disk.

,Cross-referance' datoteka u sebi sadrži podatke koji se mogu interpretirati posebnim programom CREF koji se takođe dobija uz MASM. Njegov izlaz je sortirana lista svih simbola koji se pojavljuju u asemblerskom programu zajedno sa odgovarajućim brojevima linija što može biti veoma korisno kod analiziranja u velikim programima.

Parametri koje zahteva MASM mogu se zadati i direktno u komandnoj liniji na sledeći način
C>MASM DEMO.ASM DEMO.OBJ NUL.LST NUL.CRF
ali i daleko kraće
C>MASM DEMO

Ovakvim načinom se zadaje samo naziv izvornog programa dok svi ostali parametri dobijaju inicijalne vrednosti (DEMO.OBJ NUL.LST NUL.CRF)

MASM prihvata i niz parametara u komandnoj liniji koji započinju kosom crtom. Njihova vrednost i značenje dati su u sledećoj tabeli

MASM opcioni parametri

/A	Raspoređuje po azbučnom redu
/Broj	Postavlja veličinu bafera za izvorni program na broj kilobajta
/C	Forsira kreiranje CRF datoteke i uključuje u LST datoteku linijske brojeve
/D	Formira listing u oba prolaza asembliranja. Određeni broj grešaka u prevođenju nastaje zbog različito izračunatih adresa između dva prolaza. Ovom opcijom se greške lako pronalaze
/Dsim	Definiše *sim* kao prazan string. Pošto asembler pruža mogućnost uslovnog asembliranja na sadržaj nekog simbola na konačnu formu programa ovom opcijom se može uticati i iz komandne linija bez izmene programa
/E	Asemblira za 8087 ili 80287 emulator upotrebljavajući format realnih brojeva
/Istaza	Određuje *stazu* za traženje INCLUDE datoteka. U program se naredbom INCLUDE može uključiti i više drugih izvornih asemblerskih programa ili modula. Ako se ne nalaze u tekućem direktoriju /I opcijom se može zadati gde se mogu naći
/L	Forsira kreiranje listinga izvornog programa
/ML	Čuva različitost velikih i malih slova u svim simbolima. Simboli istog naziva napisani različitim kombinacijama velikih i malih slova tretiraju se kao različiti simboli
/MX	Čuva različitost velikih i malih slova samo u PUBLIC ili EXTERN simbolima
/MU	U svim simbolima zamenjuje mala slova velikim
/N	Sprečava kreiranja liste simbola MACRO definicija itd na kraju listinga
/P	Proverava prisustvo nedozvoljenih instrukcija u 80287 zaštićenom modu
/R	Asemblira za 8087/80287 numarički koprocesor
/S	Raspoređuje segmente prema redosledu pojave u programu
/T	Sprečava ispisivanja svih poruka u toku prevođenja osim poruka o greškama
/V	Izdaje report o broju linija i simbola na kraju asembliranja
/X	U listing uključuje i delove uslovnog asembliranja koji imaju FALSE stanje
/Z	Pored poruka ispisuje i linije izvornog programa koje sadrže grešku

Svi primeri koje dajemo u ovom umetku prevođeni su IBM MACRO asemblerom. Da biste mogli da proverite kako radi i bez posebnog poznavanja MASM a dajemo neku vrstu obrasca

```
ULAZ    AH=56H
        DS=SEG (STARI NAZIV)
        DX=OFS (STARI NAZIV)
        ES=DEG (NOVI NAZIV)
        DI=OFS (NOVI NAZIV)
IZLAZ   —
GREŠKE  2 — 3 — 5 — 17
```

Funkcija je pored uobičajene promene naziva u stanju i da premesti datoteku iz jednog poddirektorija u drugi. Dovoljno je u novom nazivu upotrebiti drugu stazu. Pošto se ne vrši nikakvo kopiranje sadržaja datoteka, oba naziva moraju biti deklarisana za isti disk

Primer Promena naziva i premeštanje u drugi poddirektorij

```
CODE    SEGMENT
        MOV     DX,OFFSET STNAZ
        MOV     DI OFFSET NOVNAZ
        PUSH    DS
        POP     ES
        MOV     AH 56H
        INT     21H
        JC      ERROR

DATA    SEGMENT
STNAZ   DB      'C /TEXT/GL1 TXT',0
NOVNAZ  DB      'C /TEMP/ UVOD 001' 0
```

Slika 22

57H ČITA/MENJA VREME NASTANKA DATOTEKE
GET/SET FILE DATE AND TIME

```
ULAZ    AH=57H
        BX=FH
        ZA ČITANJE  AL=00H
        ZA PROMENU  AL=01H
                    CX=VREME
                    DX=DATUM
IZLAZ   ZA ČITANJE  CX=VREME
                    DX=DATUM
GREŠKE  1 — 6
```

Podaci o vremenu nastanka datoteke zapisani su u direktoriju diska u četiri bajta koristeći polja bitova za pojedine vrednosti na sledeći način

```
VREME   15  BIT  0
        00000 000000 00000
        (Sati) (Minuti) (Sek x 2)
        0—23 0—59 0—29

DATUM   15  BIT  0
        0000000 0000 00000
        (God-1980)(Mesec) (Dan)
        0—12 0—31
```

U istom formatu se konstruišu i vrednosti CX i DX registara

Kontrola CTRL-C interapta

Preuzimanje kontrole u slučaju ovog interapta je jedan od važnih elemenata pri pisanju velikog broja programa. Iako DOS poseduje opciju za kontrolu stanja provera ovog interapta kontrola se svodi samo na proširivanje dejstva prekida i pri obraćanju disku. Nema drugog načina da se spreči prekid programa nego da napiše sopstvena rutina za ovu namenu

Ovakva rutina može da izabere između nekoliko mogućnosti

BAJT	PODATAK
0—20	Rezervisano područje
21	Atribut datoteke
22—23	Vreme nastanka datoteke
24—25	Datum nastanka datoteke
26—29	Broj bajtova u datoteci (Četvorobajtni broj: 26 je najniži bajt, 29 najviši)
30—42	Pun naziv datoteke sa produžetkom

Funkcija podrazumeva da je prethodno postavljena adresa DTA funkcijom 1AH.
Atribut koji se zadaje u pretraživanju direktorije koristi se po sledećem algoritmu:
— ako je 0, odgovaraće sve standardne datoteke,
— ako je postavljen LABEL bit, odgovaraće samo globalni naziv diskete (VOLUME LABEL — ako postoji),
— za svaku drugu kombinaciju atribut bitova smatra se da odgovaraju sve datoteke koje imaju taj atribut i sve standardne datoteke kao kad je CX=0

Primer: Ispisivanje naziva datoteka u aktivnom direktoriju

```
CODE    SEGMENT
        MOV     DX,OFFSET DTA        ;
        MOV     AH,1AH               ;
        INT     21H                  ; postavlja DTA
        MOV     CX,110111B           ; atribut — sve sem VOLUME
        MOV     DX,OFFSET NAZ        ;
        MOV     AH,4EH               ;
        INT     21H                  ; pronalazi prvi
        JC      DONE                 ; greška ili nijedan ne odgovara
NEXT:   MOV     BX,OFFSET FNAM       ;
PO:     MOV     DL,BYTE PTR [BX]     ; ispisuje znak po znak naziva
        OR      DL,DL                ; do nule
        JZ      P1                   ;
        MOV     AH,2                 ;
        INT     21H                  ;
        INC     BX                   ;
        JMP     SHORT PO             ;
P1:     MOV     DX,OFFSET NEWLIN     ;
        MOV     AH,09H               ; dodaje novi red
        INT     21H                  ;
        MOV     AH,4FH               ;
        INT     21H                  ; sledeći naziv
        JNC     NEXT                 ; postoji?
DONE:

DATA    SEGMENT
NAZ     DB      "*.*",0
DTA     DB      30 DUP (0)
FNAM    DB      13 DUP (0)
NEWLIN  DB      13,10,'$'
```

Slika 21

54H STANJE OPCIJE ZA PROVERU ZAPISA
GET VERIFY FLAG

ULAZ AH=54H
IZLAZ AL =00H — PROVERA ISKLJUČENA
=01H — PROVERA UKLJUČENA
GREŠKE —

56H MENJA NAZIV DATOTEKE
RENAME FILE

standardnih delova asemblerskog programa. Obrazac je dovoljno uopšten da se može koristiti za najveći broj programa koje budete sami pisali — samo kod posebnih zahteva ćete morati da unosite značajnije izmene.

```
;
; KOMENTAR
;
CODE    SEGMENT PUBLIC 'CODE'
        ASSUME CS:CODE, DS:DATA, SS:STACK
DEMO    PROC    FAR
        MOV     AX,DATA
        MOV     DS,AX



EXITOK: MOV     AL,0
EXITER: MOV     AH,4CH
        INT     21H
DEMO    ENDP
CODE    ENDS
DATA    SEGMENT PUBLIC 'DATA'


DATA    ENDS
STACK   SEGMENT STACK 'STACK'
        DW      1024 DUP (?)
STACK   ENDS
```

Slika 2

Primeri će biti dati sa zaglavljima CODE SEGMENT i DATA SEGMENT, u svaki od ovih delova treba umesto tačkica uneti odgovarajući kod. U CODE segmentu smo dodeli i deo koda za izlazak iz programa. Skokom na EXITOK program prekida rad, vraćajući DOS-u status 0. Ako program prekidate usled neke greške, možete dodati ispis odgovarajuće poruke, kod greške preneti u AL registar i izaći iz programa skokom na EXITER. O ovom mehanizmu će biti detaljnije reči u pregledu DOS funkcija.

Link

Rezultat rada svakog kompajlera pa i MASM je datoteka sa takozvanim objektnim kodom (.OBJ) koja mora proći kroz LINK program da bi prešla u konačnu izvršnu verziju. Potreba za LINK-om se ne može objasniti na ovako jednostavnim primerima koje ćemo ovde davati. Ipak ćemo se osvrnuti na njegove elementarne mogućnosti.

Ako ste, koristeći prethodni obrazac, napisali program UPCASE koji sva slova nekog stringa (obeleženog labelom WRKSTR) prevodi u velika, onda bi se takav program mogao iskoristiti u velikom broju drugih programa. Potrebno je samo da umesto predloženog načina izlaska iz programa postavite RET instrukciju kako bi program mogao biti tretiran kao potprogram i da na njegovom početku dodate deklaraciju

PUBLIC UPCASE WRKSTR

i OBJ datoteka koja nastane nakon MASM moći će uz pomoć linkera da se uključi u druge programe. Primer početka jednog takvog programa bi izgledao ovako:

```
EXTRN   UPCASE:FAR,WRKSTR:BYTE
CODE    SEGMENT PUBLIC 'CODE'
        ASSUME CS:CODE, DS:DATA, SS:STACK
TEST    PROC    FAR
        MOV     AX,DATA
        MOV     DS,AX
        COPY    MSG,WRKSTR,20
        CALL    UPCASE
        COPY    WRKSTR,MSG,20
```

Slika 3

Deklaracijom EXTERN smo označili da se navedene rutine i varijable nalaze u nekom drugom OBJ modulu. (Napomena: primer zbog ilustrativnosti ima nedostatak — potrebno je svaki string nad kojim se vrši konverzija prethodno premestiti u varijablu WRKSTR iz UPCASE modula i kasnije vratiti nazad. Daleko je jednostavnije i bolje modulu proslediti samo adresu stringa koji se konvertuje. COPY nije naredba mikroprocesora već naš MACRO koji se jednostavno može kreirati na početku asemblerskog programa na sledeći način.

```
MACRO   COPY    FROM TO LEN
        MOV     SI OFFSET FROM
        MOV     DI OFFSET TO
        PUSH    DS
        POP     ES
        CLD
        MOV     CX LEN
REP     MOVSB
        ENDM
```

Slika 4

LINK proces teče na sledeći način

```
C>LINK
Microsoft (R) 8086 Object Linker Version 3 05
Copyright (C) Microsoft Corp 1983 1984 1985 All          Rights reserved
Object Modules [ OBJ] DEMO
Run File [DEMO EXE]
List File [NUL MAP]
Libraries [ LIB]
```

Slika 5

Za način unošenja parametara i prihvatanja inicijalnih vrednosti važe iste napomene kao za MASM. Isto važi i za direktno unošenje parametara u komandnoj liniji. Najčešće je u praksi dovoljno

C>LINK DEMO

Ranije navedeni primer povezivanja dve modula bi se izvodio na sledeći način

```
Object Modules [ OBJ] TEST+
Object Modules [ OBJ] UPCASE
Run File [TEST EXE]
```

Slika 6

LINK raspoznaje i određeni broj opcionih parametara koji počinju kosom crtom

Link opcioni parametri (verzija 3.05)

/C n	Postavlja maksimalno potrebnu količinu memorije pri startovanju programa na n' paragrafa. Inicijalno je n=65535 (1M9)
/D	Sve podatke iz grupe DGROUP smešta na kraj grupa
/DO	Raspoređuje segmente po MS-DOS konvenciji — prvo svi iz klase CODE zatim sve koji ne pripadaju grupi DGROUP na kraju svi segmenti iz DGROUP
/E	Sažima EXE datoteku u kondenzovani format koji odlikuje kraći kod i brže učitavanja
/HE	Ispisuje podsetnik svih opcija na ekranu
/H	Naznačava MS-DOS program loaderu da program upiše na najvišu slobodnu adresu u memoriji
/LI	U MAP datoteku uključuje i linijske brojeve izvornog programa kao pripramu za rad simboličkog dibagera (SYMDEB program)
/M	Na kraj MAP datoteke dodaje listu svih PUBLIC simbola sa njihovim vrednostima i adresama u EXE datoteci
/NOD	Ignoriše standardnu biblioteku funkcija. Svi Majkrosoftovi kompajleri u OBJ datoteku uključuju i naziv standardne biblioteke funkcija za dotični kompajler. Ovom opcijom se zadati naziv ignoriše
/NOI	Pravi razliku između istih simbola napisanih različitom kombinacijom velikih i malih slova
/NOG	Ignoriše povezanost segmenata u grupe pri dodeli adresa
/O n	Postavlja broj interapt vektora koji će koristiti OVERLAY menadžer (inicijalno 03FH)
/P	Pravi pauzu pre zapisivanja EXE datoteke da bi bila moguća izmena disketa
/SE n	Postavlja maksimalan broj segmenata u konačnom programu na vrednost n (inicijalno n=128)
/ST n	Ignoriše definiciju veličine steka iz programa i odvaja za stek n bajtova

Exe2bin

U nazivu ovog programa Majkrosoft se poigrao sa identičnim izgovorom broja 2 (TWO) i reči TO a smisao je prevedi EXE u BIN. O razlici između EXE i COM programa će biti kasnije reči. Za sada da kažemo da se ovom rutinom EXE program može prevesti u COM

- ako nema deklaracija STACK segmenta
- ako nema više od jedne deklaracije segmenta
- ako je program kraći od 64K
- ako je napisan sa ORG adresom 100H i
- ako je prvi bajt programa ujedno i prva instrukcija programa

4CH PREKIDA PROGRAM I ODREĐUJE IZLAZNI STATUS
TERMINATE WITH RETURN CODE

```
ULAZ    AH=4CH
        AL=IZLAZNI STATUS
IZLAZ   —
GREŠKE  —
```

Bez obzira na više mogućih načina za prestanak rada programa ovaj se smatra jedino preporučljivim. Razloga ima više — na zavisi od stanja segmentnih registara omogućava da se njegov izlazni status prenese u DOS varijablu ERRORLEVEL čiji se sadržaj može ispitivati unutar BAT procesa itd (vidi i objašnjenje funkcija 4DH). Po prestanku rada programa DOS propisno zatvara sve datoteke koje je program otvorio a nije izveo njihova zatvaranja. Vrednost izlaznog statusa svaki program može postaviti prema svojim konvencijama. Uobičajeno je jedino da nula označava status OK

4DH IZLAZNI STATUS POZVANOG PROGRAMA
GET RETURN CODE

```
ULAZ    AH=4DH
IZLAZ   AH=KAKO JE PROGRAM ZAVRŠIO RAD
          =00 NORMALNO
          =01 PREKINUT CTRL—C TASTEROM
          =02 PREKINUT USLED KRITIČNE GREŠKE
          =03 IZLAZ FUNKCIJOM 31H
        AL=IZLAZNI STATUS PROGRAMA
GREŠKE  —
```

Svaki program (roditelj) nakon što program koga je pozvao (dete) završi rad može uz pomoć ove funkcije saznati njegov izlazni status. Pri tome je neophodno da program „dete" prekine rad funkcijom 4CH ili (31H) postavljajući svoj izlazni status u registar AL. Ako dete koristi druge metode izlaska i program roditelj će biti prekinut što svakako nije poželjno. Zato se od verzije DOS 2 0 propisnim prestankom rada programa smatra isključivo prestanak rada upotrebom jedne od navedene dve funkcije

4EH PRONALAZI PRVI NAZIV KOJI ODGOVARA ZADATOM
SEARCH FOR FIRST MATCH

```
ULAZ    AH=4EH
        CX=ATRIBUT DATOTEKE
        DS=SEG (NAZIV)
        DX=OFS (NAZIV)
IZLAZ   DTA POPUNJEN
GREŠKE  2—18
```

4FH PRONALAZI SLEDEĆI NAZIV U DIREKTORIJU
SEARCH FOR NEXT MATCH

```
ULAZ    PRETHODNO OBAVEZAN POZIV FUNKCIJE 4EH
        AH=4FH
IZLAZ   DTA POPUNJEN
GREŠKE  18
```

Ove dve funkcije se kao i njihovi stariji ekvivalenti koriste isključivo zajedno. Prva (4E) ima zadatak da na osnovu zadatog naziva koji može sadržati i džoker znaka (? *) pronađe u direktorijumu prvu datoteku čiji naziv (i atribut) odgovaraju zadatom. Ujedno pripremа teren za funkciju 4FH kojom se nakon toga mogu dobiji i ostali nazivi koji odgovaraju. Podaci o datoteci se vraćaju tako što DOS popunjava DTA područje na sledeći način

ULAZ AH=4BH
AL=00H
ES=SEG (PARAMETAR BLOK)
BX=OFS (PARAMETAR BLOK)
DS=SEG (NAZIV PROGRAMA)
DX=OFS (NAZIV PROGRAMA)
IZLAZ SADRŽAJ SVIH(!!) REGISTARA (SEM CS I IP) UNIŠTEN
GREŠKE 1 — 2 — 5 — 8 — 10 — 11

Iz bilo kog programa moguće je izvesti bilo koji drugi program u neograničenom broju nivoa ukoliko su ispunjeni sledeći uslovi i pravilno postavljeni parametri

1 DOS mora pre poziva da za program kreira PSP Za to su mu potrebni podaci koji se postavljaju u PARAMETAR BLOK koji treba da ima sledeću strukturu

BYTE SADRŽAJ
0 — 1 — SEG (ENVIRONMENT)
2 — 3 — OFS (PARAMETRI KOJI SE PRENOSE PREKO KOMANDNE LINIJE)
4 — 5 — SEG (—)
6 — 7 OFS (prvog FCB koji ide u PSP)
8 — 9 — SEG (—)
10—11 — OFS (drugog FCB koji ide u PSP)
12—13 — SEG (—)

ENVIRONMENT blok treba da sadrži podatke u uobičajenom formatu i mora biti postavljen na granicu paragrafa (zadaje se samo segmentna vrednost adresa podrazumeva se ofset 0) Ako se za segment postavi vrednost nula program nasleđuje blok svog roditelja

Parametri koji se prenose programu preko komandne linije su takođe u formatu u koma se inače nalaze u PSP (bajt za dužinu stringa pa string koji završava se kodom 13)

2 Program roditelj mora pre poziva novog programa da oslobodi svu memoriju u sistemu koju u tom trenutku ne zauzima (funkcijom 4AH) Pošto su po povratku u glavni program svi registri uništeni neophodno je prethodno sačuvati sadržaj SS i SP registara i to tako da budu dostupni unutar CODE segmenta

Program koji će biti izveden nasleđuje sve FH otvorenih datototeka Takođe će i svaka eventualna redirekcija standardnog ulaza/izlaza biti nasleđena

Primer Poziv command com iz programa

Ovaj primer je posebno interesantan jer omogućava da se u bilo koji program ugradi jednostavna mogućnost za njegovo privremeno napuštanje i prelazak u stanje koje je identično standardnom komandnom režimu DOS Na ekranu će se pojaviti uobičajen prompt i biće moguće izvođenje DOS naredbi ili drugih programa Kada se zada interne DOS naredba EXIT kontrola se vraća našem programu Zbog ograničenosti prostora primer podrazumeva da se DOS disketa sa COMMAND COM programom nalazi u disk jedinici A



Slika 29

Prevođenja u COM se izvodi jednostavno se
C>EXE2BIN TEST EXE TEST COM

Lovci na greške

Postoji veliki broj programa koji mogu korisno poslužiti u fazi razvoja i testiranja Među prvima su dva koja posebno preporučujemo LIBRARY MANAGER — LIB kojim se mogu kreirati i održavati biblioteke standardnih rutina i jednostavno u LINK proceau uključivati u glavni program i jedan od dibagera DEBUG koji se nalazi na svakoj DOS disketi simbolički dibagar SYMDEB koji se posebno nabavlja ili najjači FULL SCREEN DEBUGER — FSD Za razvoj veoma složenih programa u alat treba uključiti i MAKE koji izuzetno pojednostavljuje i ubrzava čitav postupak prevođenja programa

Inicijalizacija DOS-a

DOS se na disketi nalazi u dva datoteka — IO SYS i MSDOS SYS koja su obeležene između ostalog i atributom skrivene tako da se ne pojavljuju pri standardnom ispisivanju sadržaja diska naredbom DIR Kada BIOS završi svoju inicijalizaciju poziva rutinu koja se takođe u njemu nalazi i naziva se „bootstrap loader" Njen zadatak je da na neko mesto u memoriji prenese sadržaj prvog sektora diska koji na svakoj formatiranoj disketi sadrži sličan programski modul zadužen za inicijalno upisivanje operativnog sistema Njegov prvi zadatak je da provari da li se na disketi nalaze dva pomenuta datoteka i da zatim upiše i startuje prvu od njih Dalji tok je veoma složen i sadrži veliki broj radnji IO SYS se sastoji iz više delova koji inicijalizuju druge i vrše njihovo premeštanje na konačna mesta u memoriji uključujući ju i MSDOS SYS modul koji je stvarno srce operativnog sistema Opis njegova inicijalizacija bi takođe zahtevao poduži spisak pa ćemo se zadovoljiti opisom samo poslednjeg koraka — startovanja COMMAND COM programa

Iako se ovaj program nalazi na svakoj sistemskoj disketi on ne spada u operativni sistem Radi se o COM programu koji je identičan svakom drugom ali je zadužen za neke važne operacije Zato je podeljen u tri dela Prvi deo stalno ostaje u memoriji obrađuje CTRL-C interapt i interapt kritična greške i ujedno obezbeđuje izlazak iz drugih programa i povratak na poznati DOS prompt Drugi deo se pojavljuje samo pri prvom upisivanju i odgovoran je za obradu AUTOEXEC BAT datoteka Nakon toga nije više potreban, pa se odbacuje Treći deo se smešta na sam vrh memorije i u sebi sadrži obrade internih DOS naredbi (DIR REN itd) startovanja programa i obradu BAT datoteka Na poziciji na kojoj se nalazi lako može biti prebrisan od strane programa koga je startovao Stalni deo (prvi pomenuti) COMMAND COM se stara da nakon prestanka rada programa se diska ponovo učita ovaj deo ako je uništen

Iako je ovako tesno povezan sa operativnim sistemom COMMAND COM u sebi ne sadrži ništa što je potrebno za rad drugih programa Mogao bi da ga zameni svaki program koji bi zadovoljio zahteve koje obavlja prvi deo COMMAND COM-a Ova pogodnost omogućuje kreiranje sopstvenog interpretera naredbi koji bi odgovarajućom napomenom u CONFIG SYS datoteci zamenio postojeći

DOS i mašinski jezik

Čitava organizacija operativnog sistema ako se uzme u obzir i organizacija zapisa na diskova i komunikacija sa periferijama može se smatrati složenom Ipak, korišćenje operativnog sistema u pisanju mašinskih programa ne zahteva poznavanje svih ovih elemenata Ono što je programeru u praksi potrebno može se sažeti u dva glavna celina (1) grupa funkcija koja nudi operativni sistem sa sistemom reagovanja na greške i (2) grupa struktura podataka koja je potrebno poznavati

Funkcija operativnog sistema se mogu lako uporediti sa naredbama bilo kog programskog jezika Ako je potrebno otvoriti datoteku u bejziku se koristi naredba OPEN Istu funkciju ima i DOS ali se jasno razlikuje način pozivanja i zadavanja parametara — naziva datoteka i slično Način pozivanja DOS funkcija je standardizovan u sledećem obliku

```
MOV   AH broj funkcija
MOV   ostali registri parametri
INT   21H
```

Veoma mali broj funkcija uglavnom onih ranijeg datuma zahtevaju da se ulazni parametri prenose na drugačiji način uglavnom popunjavanjem neke od struktura podataka

Interapt broj 21H u sebi objedinjuje prako 90 funkcija (od verzije 3 00) Pored njega koristi se i INT 20H (prekid programa) INT 25H (direktno čitanje sektora diska) INT 26H (direktno zapisivanje na sektora diska) i INT 27H (prekid programa sa permanentnim zadržavanjem u memoriji) Prvi i poslednji su u novijim verzijama zamenjenim odgovarajućim funkcijama u okviru INT 21H koje pružaju i dodatne pogodnosti tako da se sve rede koriste 25H i 26H predstavljaju gotovo direktan interfejs prema odgovarajućim BIOS rutinama Kada su ova funkcije niskog nivoa neophodno potrebne obično se direktno koristi BIOS

U DOS sistemu za programera imaju važnost još dva interapta iako se ne pozivaju iz programa To su INT 23H koji DOS poziva kada detektuje pritisak na CTRL-C ili CTRL + BREAK taster Sličan je i INT 24H — interapt na kritičnu grešku — koji se poziva kada nastane neka od alarmantnih situacija za DOS obraćanja disk jedinici u kojoj nema disketa lapis na štampač koji nije uključen i slično Ova dva interapta imaju veoma naugodne posledice u programima koji žele da imaju totalnu kontrolu nad računarom prvi će jednostavno prekinuti program dok će drugi ispisati poznatu poruku (Abort Retry Ignore?) i u slučaju da korisnik odgovori sa A (Abort) jakoda

prekinuti program DOS obezbeđuje mehanizme kojima se dosta pojednostavljeno može preuzeti kontrola nad ova dva interapta, iako rutine koje na sebe preuzimaju obradu ovih situacija mogu biti i dosta složene. O ovome će biti reči nešto kasnije.

DOS koristi više struktura podataka pri čemu su najvažnije:

PSP — PROGRAM SEGMENT PREFIX — zaglavlje programa u memoriji računara
DTA — DISK TRANSFER AREA — područje kao međustanica za prenos podataka između diska i programskih varijabli; koristi se i za neke druge namene
FCB — FILE CONTROL BLOCK — niz parametara potrebnih za otvaranje i rad sa datotekama. FCB se danas retko koristi jer od verzije 2.0 postoje efikasniji i jednostavniji metodi

Ovome se može pridodati i čitav niz različitih struktura koje se koriste u organizaciji zapisa na diskovima ali je njihovo poznavanje neophodno samo kada se pišu specijalni programi sistemskog tipa što prelazi interesovanja ovog dodatka.

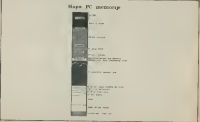
Poznavanje pobrojanih elemenata je dovoljno da se u potpunosti pokriju sledeće oblasti:

- startovanje programa, pozivanje programa iz programa i prenos parametara iz komandne linije DOS-a u program
- odvajanje potrebne količine memorije
- preuzimanje kontrole nad događajima kao što su kritične greške i prekidi CTRL-C tasterima
- kompletan rad sa datotekama — kreiranje, čitanje, zapisivanje sekvencijalnih datoteka i datoteka sa direktnim pristupom proizvoljne dužine sloga
- kompletan rad sa ostalim periferним uređajima (uz izuzetak serijskog interfejsa koji nije na propisan način podržan DOS funkcijama) i
- globalni rad sa diskovima — promena naziva, čitanje direktorija i slično

Ove mogućnosti, očigledno, pokrivaju veliki broj slučajeva — od sistemskih programa do najrazličitijih aplikacija. Treba reći da korišćenje DOS funkcija nije ograničeno samo na asemblerske programe. Gotovo svi viši jezici imaju kakav-takav direktan pristup DOS-u bez potrebe za asemblerskim modulima. Ovo se posebno odnosi na IBM PASCAL, TURBO PASCAL i C.

Odvajanje memorije

Namena pojedinih memorijskih područja je samo u RAM delu pod nadzorom DOS-a. Mnogo toga je već unapred određeno hardverom, a ROM područje je uglavnom podeljeno još u fazi konstruisanja PC računara. Bez obzira što striktno po tematici ne spada u ovaj dodatak dajemo raspored pojedinih oblasti sa nekim početnim adresama. Iako bi ovo u principu trebalo da služi samo kao orijentacija (ako se želi napisati program prenosiv na druge računare pod MS-DOS-om) ponekad može biti i od značaja pri pisanju programa.



Slika 7

Sa slike se vidi da RAM područje zauzima najviše do 640KB i da je prvi deo zauzet interapt vektorima i operativnim sistemom. U zavisnosti od verzije DOS-a prva slobodna lokacija počinje negde od 40 tog do 60 kilobajta. Od te tačke do kraja RAM područja apsolutno prostale svako

Korišćenje ove funkcije kao i ceo sistem odvajanja i oslobađanje memorije može u sadašnjim verzijama DOS delovati napotrebno. COM programi po startovanju već poseduju svu raspoloživu memoriju. EXE programi imaju mogućnost da naglase koliko im memorije treba po startovanju (zadaje se u okviru LINK procesa) ali je inicijalna vrednost, koju malo ko menja, OFFFFH paragrafa, tako da i oni imaju kontrolu nad svim memorijskim resursima sistema. Međutim oslanjanje na ovakvu organizaciju može izazvati mnogo problema. Ako zanemarimo činjenicu da će programi teže biti prenosivi na buduće verzije DOS-a koje će omogućavati rad više programe istovremeno, ostaje da program mora sam da vodi računa o adresama i dužini podataka. Daleko je jednostavnije odmah po startovanju programa (COM) osloboditi svu memoriju koja u tom trenutku nije potrebna (vidi funkciju 4AH) i onda prema potrebama zahtevati i oslobađati različite količine. Pri tome DOS vodi računa o njihovom raspoređivanju. Na programu ostaje samo da koristi početnu adresu bloka koju mu obezbeđuje DOS.

Ako program privremeno zahteva svu raspoloživu memoriju, obično se koristi procedura koja je data u sledećem primeru.

Primer: preuzimanje čitave raspoložive memorije

```
CODE      SEGMENT
          MOV      AH 48H
          MOV      BX,OFFFFH                 ,zahtev za 1MB sigurno
          INT      21H                       ,ne može biti ispunjen pa
          JNC      nemoguće                   DOS vraća CY=1,
          MOV      AH 48H                     BX=maksimalna količina
          INT      21H
          JC       nemoguće                  ,zahtev sada mora biti zadovoljen
          MOV      BUF_SEG,AX                ,čuva segment adrese bloka

DATA      SEGMENT
BUF SEG   DW       0
```

Slika 10

49H OSLOBAĐA DODELJENU MEMORIJU

RELEASE MEMORY

ULAZ AH=49H
ES=SEG (ADRESA BLOKA)
IZLAZ —
GREŠKE 7—9

Blok odvojen funkcijom 48H vraća se sistemu i program ga više ne može koristiti. ES mora sadržati istu vrednost koju je funkcija 48H vratila u registru AX.

4AH MENJA KOLIČINU DODELJENE MEMORIJE

MODIFY MEMORY ALLOCATION

ULAZ AH=4AH
BH=NOVA VELIČINA BLOKA (U PARAGRAFIMA)
ES=SEG (ADRESA BLOKA)
IZLAZ BX=MAKSIMALNA KOLIČINA
(AKO NEMA DOVOLJNO MEMORIJE DA SE ZADOVOLJI ZAHTEV)
GREŠKE 7—8—9

Dodeljeni memorijski blok se može ili smanjiti ili povećati. Za ispravan rad ove funkcije neophodno je da je adresa u ES registru identična onoj koju je vratila funkcija 48H kada je blok odvajan. Pošto se COM programima dodeljuje sva raspoloživa memorija u trenutku startovanja ova funkcija se često koristi da se nepotreban deo oslobodi. U tom slučaju ES treba da sadrži segment adrese PSP, praktično sadržaj koji ima odmah po startovanju programa.

Primer: COM program oslobađa nepotrebnu memoriju

```
CODE      SEGMENT
          MOV      BX 0800H                  programu dovoljno 32K
          MOV      AH 4AH
          INT      21H
          JC       ERROR
```

4BH IZVRŠAVANJE PROGRAMA

EXECUTE PROGRAM

41H BRISANJE DATOTEKE
DELETE FILE

ULAZ: AH=41H
DS=SEG(NAZIV)
DX=OFS(NAZIV)

IZLAZ: —
GREŠKE 2—5

Funkcija za razliku od odgovarajuće ranije varzije 13H ne može obrisati više datoteka istovremeno korišćenjem džoker znakova u nazivu.

42H MENJA POZICIJU SLEDEĆEG ČITANJA/ZAPISIVANJA
MOVE FILE POINTER

ULAZ AH=42H
BX=FH
CX=VIŠA REČ VREDNOSTI POMERANJA
DX=NIŽA REČ VREDNOSTI POMERANJA
AL=METOD PROMENE
0 — APSOLUTNO OD POČETKA DATOTEKE
1 — RELATIVNO OD TRENUTNE POZICIJE
2 — RELATIVNO OD KRAJA DATOTEKE

IZLAZ. DX=VIŠA REČ NOVE APSOLUTNE POZICIJE U ODNOSU NA POČETAK
AX=NIŽA REČ
GREŠKE 1—6

Vrednosti za koje se pomera pozicija mogu biti pozitivne ili negativne. Pomeranje pozicije preko kraja datoteke ne proizvodi grešku. Pomeranje pozicije ispred početka datoteka proizvešće grešku tek pri kasnijem pokušaju čitanja ili zapisivanja. Funkcijom se može odrediti i veličina datoteke. Nakon pomeranja pozicije za 0 bajtova od kraja datoteke u registrima DX,AX se nalazi njena veličina.

43H ČITA/DODREĐUJE ATRIBUTE DATOTEKE
GET OR SET FILE ATTRIBUTS

ULAZ AH=43
AL=0 — ČITA ATRIBUT BAJT
1 — MENJA ATRIBUT
CX — NOVI ATRIBUT
DS=SEG(NAZIV)
DX=OFS(NAZIV)

IZLAZ CX=ATRIBUT NAKON ČITANJA
GREŠKE 1—2—3—5

47H AKTIVNI DIREKTORIJ
GET CURRENT DIRECTORY

ULAZ AH=47H
DL=BROJ DISKA (0=AKTIVAN 1=A 2=B ITD.)
DS=SEG (BUF)
SI=OFS (BUF)

IZLAZ BUF POPUNJEN PUNIM NAZIVOM AKTIVNOG DIREKTORIJA
GREŠKE 15

BUF mora imati dužinu od bar 64 bajta. Naziv aktivnog direktorije DOS upisuje u BUF sa stazom kojom se do njega stiže. Naziv ne uključuje slovo disk jedinice ni vodeću obrnutu kosu crtu a završava bajtom 0. Ako je osnovni direktorij aktivan BUF će sadržati samo 0 bajt.

48H ODVAJA MEMORIJSKI BLOK
ALLOCATE MEMORY

ULAZ AH=48H
BX=VELIČINA BLOKA (U PARAGRAFIMA)

IZLAZ BX= MAKSIMALNA KOLIČINA
(AKO NEMA DOVOLJNO MEMORIJE DA SE ZADOVOLJI ZAHTEV)

GREŠKE 7—8

interesovanje programere za stvarne memorijske lokacije. U DOS je ugrađen potpun mehanizam za upravljanje raspodelom ovog područja (MEMORY ALLOCATION) tako da neme potrebe za vođenjem računa o apsolutnim adresama na koje će se smeštati program i podaci. To je zapravo gotovo i nemoguće jer takav red retko može završiti normalnim prekidom programa i povratkom u DOS.

Da bi obezbedio kontrolisanu raspodelu memorije DOS RAM područje deli na takozvane memorijske arene. Svaka arena se sastoji iz zaglavlja i samog memorijskog područja koje je dodeljeno programu. Zaglavlje sadrži pokazivač na sledeće čineći tako ulančanu listu. Zahvaljujući ovom sistemu DOS je u stanju da na zahtev programa odvoji od slobodne količine traženu količinu prosleđujući programu početnu adresu područja. da područje „oslobodi" i vrati u oblast slobodne memorije bez potrebe za ikakvim premeštanjem po memoriji već samo održavanjem liste.

Količina memorije koja se može dodeliti programima se kreće od 1 paragrafa (16 bajtova) do maksimalne slobodne količine u trenutku zahteva. Sistem upravljanja raspodelom memorije može da nametne DOS-u mnogo problema o kojima treba da vodi računa. Pretpostavimo da neki zamišljeni program obrađuje veći broj stringova čije su dužine dovoljno velike i razlikuju se od stringa do stringa (recimo od 50 do 5000 bajtova). Ako program za svaki string koji mu stigne na obradu zatraži od DOS-a odgovarajuću količinu memorije brzo će nastati dugačka liste zaglavlja arena. Ako neki od stringova nije potreban program može osloboditi područje koje je zauzimao pa se u memoriji pojavljuju neiskorišćene „rupe" o čemu DOS takođe mora da vodi računa. Dakle zahtevi za brzinom i optimalnim korišćenjem memorije stavljaju DOS na velike muke.

Ovakvi zahtevi su često toliko protivurečni da ih je nemoguće ne zadovoljavajući način pomiriti. Zbog toga je od verzije 3.0 ugrađena mogućnost da korisnik sam zada DOS-u strategiju odvajanje memorije koja najbolje odgovara njegovoj aplikaciji. Ako se zada strategija „prvog uklapanja" (first fit) DOS će pretraživati listu od početka i prvo dovoljno veliko područje koje zadovoljava zahtev dodeliti programu. Ovom strategijom se postiže brzina ali je prali neracionalno korišćenje memorije jer prvo slobodno područje može biti i mnogo veće od zahtevanog. Strategijom „najboljeg uklapanja" (best fit) uvek se pronalazi najmanje slobodno područje dovoljno veliko da zadovolji zahtev čime se memorije optimalno koristi ali se brzina rada smanjuje.

Direktnim pristom memoriji upotrebom proizvoljno izabranih adresa može se desiti da se uništi lista zaglavlje što će najverovatnije sprečiti korektan prestanak rada programa. Slično se dešava i pri nepažljivom korišćenju područja koje je DOS dodelio programu — pisanje preko dozvoljene adresne granice gotovo sigurno ima isti efekat.

Razumevanje i poštovanje načina odvajanja memorije je jedan od presudnih faktora pri pisanju korektnih programa. Treba reći da i DOS kada treba da odvoji memoriju za upis i startovanje programa koristi iste funkcije i mogućnosti koje su dostupne i programeru.

COM i EXE programi

Imajući u vidu ono što je do sada rečeno o odvajanju memorije lako se može pokazati kako se odvija tipičan tok upisivanja i startovanja programa. Dovoljno je upoznati samo još jednu važnu strukturu podataka koja se u DOS terminologiji naziva PROGRAM SEGMENT PREFIX (u daljem tekstu PSP).

PSP predstavlja zaglavlje svakog programa koje DOS kreira od prve. u tom trenutku slobodne memorijske lokacije i u sebi sadrži više informacija koje su potrebne kako programu tako i operativnom sistemu. Kao primer ćemo upotrebiti neki program TEST.COM koji je startovan na sledeći način

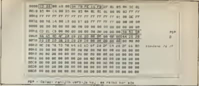
C>PSPDEMO PSPDEMO /D /F

Za njega će DOS u memoriji kreirati sledeći PSP (uvek se kreira na granici paragrafa. Ime ofset adrese 0)

```
0000 CD 20 00 A0 00 9A F0 FE 1D F0 2F 01 B5 0A 2C 01
0010 B5 0A C4 04 B5 0A B5 0A 0A 01 01 00 02 03 FF FF
0020 FF FF FF FF FF FF FF FF FF FF FF FF 81 00 DE FF
0030 00 30 14 00 18 00 07 03 FF FF FF FF 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050 CD 21 CB 00 00 00 00 00 00 00 00 00 00 50 53 50
0060 44 45 4D 4F 20 20 20 20 00 00 00 00 00 44 20 20
0070 20 20 20 20 20 20 20 20 00 00 00 00 00 00 00 00
0080 0E 20 70 73 70 64 65 6D 6F 20 2F 64 20 2F 66 0D     pspdemo
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Slika 8

Veliki deo podataka koji se nalaze u PSP je ostatak od ranijih verzija DOS, pa ćemo ih samo izolovano pokazati bez posebnih objašnjenja

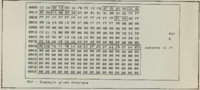


Slika P

Prva dva bajta (CD 20) predstavljaju mašinsku instrukciju INT 20H koja je u ranijim verzijama služila za prestanak rada programa. Pošto se nalazi na adresi CS:0, programi su za izlazak mogli koristiti i CALL CS:0, što direktno odgovara instrukciji CALL 0000 kojom se u CP/M u napušta program. Slično je i sa sledećom grupom od 5 bajtova koji u stvari predstavljaju instrukciju za skok na dispečer naredbi sa istim efektom kao INT 21H. U CP/M-u se koristi CALL 0005.

Sledeće grupe bajtova od ofseta 005CH predstavljaju dva neotvorena FCB (FILE CONTROL BLOCK) o kojima će biti reči kada na red dođe rad sa datotekama.

Deo koji je danas u upotrebi sadrži sledeće elemente



Slika 16

Prva dva bajta predstavljaju gornju granicu memorije dostupnu programu (u paragrafima). U konkretnom slučaju je to vrednost A000H (640K) što je ujedno i gornja fizička granica RAM memorije na računaru na koma je izvedan primer. Vrednost ne mora u svim situacijama odgovarati maksimalnoj količini RAM-a, o čemu će biti reči nešto kasnije.

Sledeće tri grupe od po četiri bajta predstavljaju redom sadržaje interapt vektora 22H, 23H i 24H koja su imali po startovanju programa. Ova tri interapta su zadužena za kraj rada programa, obradu CTRL-C prekida i kritičnih grešaka. Programi mogu jednostavno preuzimati obradu ovih događaja na sebe promenom vrednosti vektora DOS funkcijom 25H. Po prestanku rada programa DOS će, zahvaljujući tome što ih je sačuvao u PSP, vratiti vektorima originalan sadržaj. Zbog toga nipošto nije dozvoljeno menjati ova vrednosti u PSP-u.

Područje od ofseta 0080 ima posebno korisnu namenu. Ako uporedite njegov sadržaj sa sadržajem komandne linije kojom je program prozvan, primetićete da se u njemu nalazi sve što je u liniji otkucano osim naziva programa. Ovo je atikasan način za prenos parametara iz komandne linije u program i koristi se veoma često. Smatra se najpraporučljivijim metodom za programe sistemskog tipa. Sadržaj komandne linija je pranet u identičnom obliku u koma je otkucan, s tim što se na prvom mestu nalazi broj znakova u liniji (u primeru 0EH). Linije obavezno završava

ULAZ AH=3EH
BX=FH

IZLAZ —

GREŠKE 6

FH je broj sačuvan nakon funkcije OPEN za dotičnu datoteku

3FH ČITA SADRŽAJ DATOTEKE

READ FILE OR DEVICE

```
ULAZ    AH=3FH
        BX=FH
        CX=BROJ BAJTOVA KOJI SE ČITA
        DS=SEG(BUF)
        DX=OFS(BUF)
IZLAZ   AX=BROJ PROČITANIH BAJTOVA
GREŠKE  5—6
```

Iz datoteke čiji je FH dobijen nakon OPEN čita se CX bajtova i prenosi u befer čija je adresa zadata. Sledeće čitanja ide od pozicije gde je prethodno stalo. Vrednost koja se vraća u AX registru upotrebljava se za detektovanje kraja datoteka. Ako je AX<CX samo deo traženog broja bajtova je pročitan jer je naišao kraj. Ako je AX=0 datoteka je bila pročitana do kraja pri poslednjeg poziva funkcije. Funkcija se koristi i za rad sa periferním uređajima kao i sa inicijalno otvorenim FH (npr. pročitaj 20 bajtova sa standardnog ulaza — CX=20, BX=0)

Primer: Obrada sadržaja datoteke

```
CODE     SEGMENT
         MOV     DX OFFSET NAME
         MOV     AH 3DH
         MOV     AL 0
         INT     21H                    otvara za čitanja
         JC      not found
         MOV     HANDLE AX
LOOP
         MOV     BX HANDLE
         MOV     CX 4096
         MOV     DX OFFSET BUF
         MOV     AH 3F
         INT     21H                    čita po 4K
         OR      AX,AX
         JZ      EOF
                 obrada
         JMP     LOOP
EOF
         MOV     BX,HANDLE              zatvara datoteku
         MOV     AH 3EH
         INT     21H

DATA     SEGMENT
NAME     DB      /DEMO DAT' 0
HANDLE   DW      ?
BUF      DB      4096 DUP (?)
```

Slika 18

40H UPIS U DATOTEKU

WRITE FILE OR DEVICE

```
ULAZ    AH=40H
        BX=FH
        CX=BROJ BAJTOVA KOJI SE ZAPISUJE
        DS=SEG(BUF)
        DX=OFS(BUF)
IZLAZ   AX=BROJ ZAPISANIH BAJTOVA
GREŠKE  5—6
        AX=0 — DISK PUN
```

Važe iste napomene kao za funkciju 3FH

Ova funkcija realizuje internu DOS naredbu MKDIR

PRIMER — Kreiranje poddirektorija

```
CODE    SEGMENT
        MOV     DX OFFSET NAME
        MOV     AH 39H
        INT     21H
        JC      ERROR

DATA    SEGMENT
NAME    DB         C /TEXT/GAL 0
```

Slika 17

3AH UKLANJA PODDIREKTORIJ
DELETE SUBDIRECTORY

ULAZ AH=3AH
DS=SEG(NAZIV)
DX=OFS(NAZIV)
IZLAZ —
GREŠKE 3—5—6—16

Funkcija realizuje internu DOS naredbu RMDIR

3BH MENJA AKTIVAN DIREKTORIJ
SET CURRENT DIRECTORY

ULAZ. AH=3BH
DS=SEG(NAZIV)
DX=OFS(NAZIV)
IZLAZ —
GREŠKE 3

Funkcija realizuje internu DOS naredbu CHDIR

3CH KREIRA/PONIŠTAVA DATOTEKU
CREATE OR TRUNCATE FILE

ULAZ AH=3CH
CX=ATRIBUT KOJI ĆE BITI KORIŠĆEN
DS=SEG(NAZIV)
DX=OFS(NAZIV)
IZLAZ AX=FH
GREŠKE 3—4—5

Kreira se nova datoteka se zadatim nazivom ili poništava stara ako postoji FH se koristi u daljem obraćanju datoteci Datoteka je otvorena i za čitanje i za zapisivanje

3DH OTVARA DATOTEKU
OPEN FILE

ULAZ AH=3DH
AL=0 — SAMO ČITANJE
1 — SAMO ZAPISIVANJE
2 — ČITANJE/ZAPISIVANJE
DS=SEG (NAZIV)
DX=OFS (NAZIV)
IZLAZ AX=FH
GREŠKE 1—2—3—4—5—12

Početna pozicija u datoteci od koje teče upis/ispis se postavlja na nulu

3EH ZATVARA DATOTEKU
CLOSE FILE

kodom 13 (RETURN) koji se ne računa u broj znakova Treba napomenuti da se u slučaju upotrebe redirekcija standardnog ulaza/izlaza na primer

C>PSPDEMO PSPDEMO /D /F <DEMO IN >DEMO DTA

parametri redirekcija ne prenose u ovo područje Tako će sadržaj i u ovom primeru biti isti kao i u prethodnom bez redirekcija te se neće pojaviti tekst <DEMO IN >DEMO DTA Na ovaj način redirekcija je transparentna (nevidljiva) za program Pošto se smatra da će program ispitati parametre odmah na početku rada i da mu kasnije više neće biti potrebni isto područje se koristi i kao inicijalni DTA (DISK TRANSFER ADDRESS)

Dakle kada iz komandne DOS linije zadate naziv nekog programa bez obzira da li se radi o EXE ili COM verijanti DOS preduzima sledeće korake

1 odvaja količinu memorije koja je potrebna za program,
2 na dnu ova količine gradi PSP
3 prenosi sadržaj ostatka komandne linije u PSP
4 prenosi sadržaje interapt vektora 22H 23H i 24H u PSP
5 upisuje program
6 dodeljuje inicijalne vrednosti segmentnim registrima i
7 skače na prvu instrukciju programa

Bez obzira što su postupci isti način izvođenja koraka 1,5 6 i 7 se kod COM i EXE programa razlikuju Ova razlika se pojavljuje usled njihove različite strukture

COM programi su u datoteci zapisani na identičan način na koji se i prenose u memoriju Prvi bajt u datoteci je i prvi bajt prve mašinske instrukcije programa čime je ulazna adresa programa uvek 100H (prva slobodna lokacija nakon PSP na koju se program upisuje) Pošto kod programa ne prelazi 64K sve adresiranja su relativna unutar segmenta pa su samim tim nezavisne od pozicija na kojoj će se program naći u memoriji Takođe svi segmenti inicijalno imaju istu vrednost COM programi se često nazivaju i mali memorijski modeli

EXE programi mogu biti proizvoljno veliki Pošto mogu imati posebno definisane kod data i stek segmente mora se definisati njihov inicijalni sadržaj pri startovanju programa Prelazeći granicu od 64K$ svaki skok manje sadržaj na samo pointera instrukcija (IP) već i CS (kod segmenta) pa tako postaje zavisan od realnih memorijskih adresa na kojima se program nalazi Ove pak ne mogu biti unapred poznate pre samog trenutka upisivanja tako da se svaka takva instrukcija mora u EXE datoteci na neki način obeležiti kako bi se prilagodila kada se program nađe u memoriji Zbog svega toga zapis na disku ima zaglavlje u koma se nalaze sve neophodne informacije EXE programi se nazivaju i srednjim ili velikim memorijskim modelima

Razlike u načinu izvođenja pojedinih koraka pri startovanju programa ćemo sumirati sledećom tabelom

Korak	EXE	COM
1	DOS pokušava da odvoji maksimalno potrebnu količinu koju je u zaglavlja EXE datoteke postavio LINK (vidi opcioni LINK parametar /C) Ukoliko to nije moguće odvaja se onoliko koliko je na raspolaganju Ako ova količina ne zadovoljava minimum potreban programu (dužina programa+podaci+stek) ispisuje se poruke i program se ne startuje	Dodeljuje se sve raspoloživa memorija Ako je manja od dužina programa ispisuje se poruka i program se ne startuje
5	Prenosi se sadržaj EXE datoteke i uz pomoć tablice iz zaglavlja prilagođavaju one instrukcije koje zavise od apsolutnih memorijskih lokacija	Sadržaj datoteke se direktno bez izmena preslikava u memoriju
6	DS i ES se postavljaju na PSP SS prema deklaraciji segmenta sa STACK atributom SP (stek pointer) prema veličini segmenta STACK	DS ES i SS se postavljaju na PSP SP dobija vrednost OFFFEH ako je moguće Ako nema dovoljno memorije spušta se na maksimalnu granicu
7	Skok se izvodi na segment i ofset one labela koja je u asembleru označena END naredbom na kraju programa	Uvek se izvodi skok na CS 100H

Postojanje COM i EXE varijanti programa ponekad može da izazove nedoumice oko toga koji model izabrati za sopstveni program U praksi se pokazalo da je COM format veoma pogodan za pisanje kraćih programa koji su jednostavnije organizacije To se posebno odnosi na takozvane memorijski rezidentne programe koji po učitavanju ostaju trajno u memoriji i obično koriste neki interapt za povremena aktiviranja Tipični primeri su razni KEYS programi za promenu rasporeda tastature ili programi kao što je SIDEKICK COM program nema ograničenja za ukupno korišćenje memorije jedino sam kod ne može prelaziti jedan segment Programer je slobodan

da u toku programa menje sadržaj DS ES ili SS registare i pristupa proizvoljnom delu dodeljene mu memorije obrađujući velike količine podataka.

EXE format je primenjiv u gotovo svim situacijama. Nije posebno pogodan za rezidentne programe, ali je nezamenjiv kod velikih i složenih programa.

Rad sa datotekama

Datoteke se dodecime se na disku zapisuju na identičan način — prostim ređanjem podataka. U datoteci neme nikakvih zaglavlja niti drugih posebnih informacija. Podaci o svakoj datoteci su zapisani u direktoriju diska koji čuva, između ostalog, i sledeće:

1. Naziv datoteke
2. Tačnu dužinu u bajtovima. Prostor koji datoteka na disku zauzima je uvek veći jer se odvajanje vrši u takozvanim jedinicama alokacije — klasterima koji objedinjuju više fizičkih sektora diska. Datoteka sa samo jednim bajtom podataka zauzeće 1K (dva sektora) jer je to minimalna jedinica odvajanja koju koristi DOS na disketama. Informacija u direktoriju je jedini način da se sazna tačna dužina korisnih podataka.
3. Vreme i datum nastanka/izmene datoteke
4. Atribut datoteke (jedan bajt) u kome svaki bit ime sledeće značenje:

bit	atribut	
0	READ ONLY	— datoteka samo za čitanje,
1	HIDDEN	— skrivena datoteka, naziv se ne pojavljuje u direktoriju
2	SYSTEM	— sistemska datoteka,
3	VOLUME	— ulaz u direktoriju nije datoteka već globalni naziv diskete,
4	SUB—DIR	— datoteka je poddirektorijum i u sebi sadrži iste podatke za sve datoteke koje se u njemu nalaze
5	ARCHIVE	— svaka izmena sadržaja datoteka postavlja ovaj atribut, BACKUP i RESTORE naredbe ga koriste da utvrde da li je sadržaj datoteka menjan od poslednjeg pravljenja zaštitnih kopija.

DOS nudi dva pristupa podacima — sekvencijalan i direktan. Dužina sloga podataka koji se koristi u datotekama sa direktnim pristupom nigde se ne zapisuje, tako da drugi programi koji koriste iste podatke moraju ovu informaciju imati ugrađenu kao unapred poznatu veličinu. U funkcijama višeg nivoa koje se danas koriste najpreciznije bi bilo reći da DOS koristi neku vrstu mešavine ova dva pristupa, mesto se koga će sledeći podatak iz datoteke biti pročitan može se u bilo kom trenutku proizvoljno postaviti, kao što se u svakom trenutku može pročitati i proizvoljno dugačak slog.

Malu zabunu kod programera može izazvati navika rada sa datotekom iz viših programskih jezika koji, zahvaljujući ovako fleksibilnom metodu čitanja, kreiraju sopstvena klasične sekvencijalne i direktne pristupe. Isto važi i za uobičajenu podelu datoteka na ASCII sekvencijalne, kod kojih se krajem linije smatraju CR/LF karakteri (ASCII kodovi 13 i 10) a krajem datoteke EOF marker (ASCII kod 26) i binarne kod kojih je kraj datoteke fizički kraj zapisa. Ništa od ovih podela ne postoji u DOS-u. Svaka datoteka se smatra nizom bajtova određena dužina koji se mogu čitati sa proizvoljnog mesta u proizvoljnoj količini.

Treba napomenuti da se i ostali periferni uređaji tretiraju na sličan način. Otvaranje kanala za vezu se odvija kao i otvaranje datoteka uz korišćenje logičkog naziva uređaja — CON (konzola) LPT1 PRN LST (printer) COM1 AUX (RS232 interfejs) itd.

Tradicionalni način — FCB

Kao što je već napomenuto, DOS nudi dva različita načina rada sa datotekama. Pošto je prvi upotrebom FCB (FILE CONTROL BLOK) strukture već prilično zastareo i Majkrosoft ga više ne preporučuje, samo ćemo se u kratkim crtama osvrnuti na njega.

FCB predstavlja niz podataka u tačno definisanom formatu koji su neophodni za otvaranje i rad sa datotekom. Njegova osnovna varijanta je dugačka 37 bajtova i ima sledeći izgled:

OFSET	SADRŽAJ
00H	Broj disk jedinice
01H	Naziv datoteka levo poravnat
09H	Produžetak naziva datoteka levo poravnat
0CH	Redni broj bloka koji se trenutno čita
0EH	Dužina sloga
10H	Veličina datoteka u bajtovima
14H	Datum kreiranja/izmena datoteke
16H	Vreme kreiranja/izmene datoteka
18H	Rezervisano sistemsko područje
20H	Redni broj sloga
21H—24H	Redni broj sloga za datoteka sa direktnim pristupom

Postupak otvaranja datoteke teče na sledeći način:

- program odvaja 37 bajtova za FCB
- program postavlja naziv i produžetak naziva datoteka
- program poziva odgovarajuću DOS funkciju za otvaranje datoteka

31H PREKIDA PROGRAM I ZADRŽAVA GA U MEMORIJI
TERMINATE AND STAY RESIDENT (KEEP)

```
ULAZ    AH=31H
        AL=IZLAZNI KOD
        DX=POTREBNA MEMORIJA U PARAGRAFIMA
IZLAZ   —
GREŠKE  —
```

Kod programa koji trajno ostaju u memoriji (kao što su DOS-ov PRINT i slični) ovo je jedini preporučljiv metod izlaska iz programa. Izlazni kod se prenosi u ERRORLEVEL varijablu koja se može ispitivati u okviru BAT procedure. Količina memorije zadate DX registrom ostaje trajno dodeljena programu pa se preporučuje da bude minimalno potrebna. Sve otvorene datoteke će ovim izlaskom biti automatski zatvorena.

33H ČITA/ODREĐUJE STANJE PROVERE CTRL-BREAK PREKIDA
GET OR SET CTRL-BREAK FLAG

```
ULAZ    AH=33H
        AL=0 — ČITA STANJE
        AL=1 — ODREĐUJE STANJE
        DL=0 — ISKLJUČUJE PROVERU
        DL=1 — UKLJUČUJE PROVERU
IZLAZ   DL=0 — PROVERA ISKLJUČENA
        DL=1 — PROVERA UKLJUČENA
GREŠKE  —
```

Ako je stanje uključeno, svaki zahtev za upis ili ispis će prethodno proveriti da li je korisnik pritisnuo kombinaciju CTRL-BREAK (CTRL-C) u kom slučaju se izvodi INT 23H. Stanje je globalno za čitav sistem pa je propisno da program sačuva stanje pre nego što ga izmeni i pre završetka rada vrati originalnu vrednost.

35H ČITA INTERAPT VEKTOR
GET INTERRUPT VECTOR

```
ULAZ    AH=35H
        AL=BROJ INTERAPTA
IZLAZ   ES=SEG (VEKTOR)
        BX=OFS (VEKTOR)
GREŠKE  —
```

36H KOLIČINA SLOBODNOG PROSTORA NA DISKU
GET FREE DISK SPACE

```
ULAZ    AH=36H
        DL=BROJ DISK JEDINICE (0=AKTIVNA, 1=A 2=B ITD.)
IZLAZ   AX=BROJ SEKTORA KOJI ČINE KLASTER
        BX=BROJ SLOBODNIH KLASTERA
        CX=FIZIČKA VELIČINA SEKTORA
        DX=UKUPAN BROJ KLASTERA NA DISKU
GREŠKE  AX=0FFFFH — POGREŠAN BROJ DISKA
```

Ukupan slobodni prostor u bajtovima se dobije množenjem AX×BX×CX

39H KREIRA PODDIREKTORIJ
CREATE SUBDIRECTORY

```
ULAZ    AH=39H
        DS=SEG (NAZIV)
        DX=OFS (NAZIV)
IZLAZ   —
GREŠKE  3 5 6 16
```

2BH PODEŠAVA DATUM NA SISTEMSKOM KALENDARU
SET SYSTEM DATE

ULAZ AH=2AH
CX=GODINA (1980—2099)
DH=MESEC
DL=DAN
IZLAZ AL=0
GREŠKA AL=0FFH — NEKA OD VREDNOSTI VAN OPSEGA

2CH ČITA VREME SA SISTEMSKOG ČASOVNIKA
GET SYSTEM TIME

ULAZ AH=2CH
IZLAZ CH=SATI (0—23)
CL=MINUTA
DH=SEKUNDI
DL=STOTINKI
GREŠKE —

2DH PODEŠAVA VREME NA SISTEMSKOM ČASOVNIKU
SET SYSTEM TIME

ULAZ. AH=2CH
CH=SATI (0—23)
CL=MINUTA
DH=SEKUNDI
DL=STOTINKI
IZLAZ AL=0
GREŠKE AL=0FFH — NEKA OD VREDNOSTI VAN OPSEGA

2EH PROVERA ZAPISA NAKON UPISIVANJA NA DISK
SET VERIFY FLAG

ULAZ AH=2EH
AL=0 — ISKLJUČI PROVERU
=1 — UKLJUČI PROVERU
AL=0
IZLAZ —
GREŠKE —

Provera zapisa ako je uključena automatski se vrši nakon svakog zapisivanja Donekle usporava rad sistema, a u praksi se pokazala potpuno nepotrebnom

2FH ADRESA AKTIVNOG DTA
GET DTA ADDRESS

ULAZ AH=2FH
IZLAZ. ES=SEG (DTA)
BX=OFS (DTA)
GREŠKE —

Funkcija vraća adresu trenutno aktivnog DTA područja

30H VERZIJA OPERATIVNOG SISTEMA
GET MS-DOS VERSION NUMBER

ULAZ AH=30H
IZLAZ AL=0 — DOS 1 00
AL>0 — VIŠI BROJ VERZIJE (DOS 3 20=3)
AH=NIŽI BROJ VERZIJE (DOS 3 20=2)
GREŠKE

- DOS pronalazi datoteku na disku
- DOS popunjava polja sa vremenom, datumom i dužinom datoteke
- DOS postavlja dužinu sloga (ofset OEH) na inicijalnu vrednost 128

Pre otvaranja datoteke program mora da odvoji i odgovarajuće veliki deo memorije u koji će biti prenošeni podaci iz datoteke Ovo područje se naziva DTA — DISK TRANSFER AREA i odgovarajućom DOS funkcijom se DOS-u stavlja na znanje koju je početnu adresu program za ovo područje izabrao

Dalji tok rada sa datotekom zavisi od načina pristupa Ako je čitanje sekvencijalno dovoljno je pozivati DOS funkciju za sekvencijalno čitanje Podaci će u blokovima zadate veličine (ofset OEH FBC) biti prenošeni sa diska u DTA i odgovarajući pointeri u FBC inkrementirani kako bi sledeći poziv čitao nastavak datoteke Lokacija sa koje će sledeći blok biti pročitan dobija se kombinacijom vrednosti sa ofseta OCH i 20H iz FCB

Kod datoteka sa direktnim pristupom (RANDOM) pre čitanja treba postaviti dužinu sloga (OEH) ako inicijalna vrednost 128 ne odgovara Redni broj sloga koji će biti pročitan takođe mora biti postavljen od ofseta 21H u obliku četvorobajtnog broja (najniži bajt prvi) Apsolutna pozicija u datoteci sa koje će biti pročitani podaci se dobija kombinacijom ove vrednosti i dužine sloga (OEH)

Kada je u DOS uvedeno obeležavanje datoteka atributom struktura FCB je postala nedovoljna pa je uveden i prošireni FCB kod njega prvi bajt ima vrednost OFFH (što inače nikad ne može biti broj disk jedinice pa se tako razlikuje od standardnog FCB-a) zatim sleda 5 bajtova rezervisanog DOS područja i konačno atribut. Ostatak je identičan standardnom FCB Ukupna dužina proširenog FCB je 44 bajta

Očigledno je da ovakav metod rada sa datotekama ima mnogo nedostataka

1) Ne može se koristiti hijerarhijska organizacija direktorija jer se u FCB definiše samo naziv datoteke a ne i staza pa datoteka mora biti u aktivnom direktoriju
2) Metod je komplikovan zahteva mnogo predradnji
3) Pozicioniranja i čitanja blokova podataka unutar datoteke je nepraktično

Metod ipak ima nekoliko prednosti

1) Broj istovremeno otvorenih datoteka nije ograničen
2) Informacija o vremenu kreiranja i dužini datoteka dostupna su odmah po otvaranju što je jednostavnije od korišćenja posebnih funkcija za ovu namenu

Savremen način — File Handles

Za otvaranje datoteke dovoljno je zadati njen naziv koji može uključivati i stazu poddirektorija kojom se do nje stiže Na osnovu ove informacije DOS interno kreira blokova podataka potrebne za rad sa datotekama, a programu vraća mali ceo broj kojim će u toku čitanja i pisanja identifikovati ovu vezu Ovaj broj se naziva FILE HANDLE (slobodniji prevod — broj za rukovanje datotekom u daljem tekstu FH) Kada treba pročitati n bajtova iz datoteke program poziva odgovarajuću funkciju navodeći FH adresu od koje treba smeštati podatke broj bajtova i to je sve FH je obavezno ulazni parametar svih funkcija koja rade sa datotekama

Prilikom inicijalizacije DOS-a otvara se određeni broj datoteka Upotrebljeni su sledeći FH brojevi

FH=0 — CON kao ulaz — standardni ulaz
FH=1 — CON kao izlaz — standardni izlaz
FH=2 — CON kao izlaz — standardni kanal za ispis greške
FH=3 — AUX kao ulaz/izlaz — standardni pomoćni uređaj (RS232)
FH=4 — LST kao izlaz — standardni printer kanal

Za ove veze nije potrebno otvarati kanale — dovoljno je koristiti dati FH

Ovakvim metodom se može lako ostvariti preusmeravanja standardnog ulaza/izlaza Osobina FH brojeva je da se za svaku novu datoteku bira najmanji slobodan broj Ako se tako DOS funkcijom CLOSE zatvori veza sa FH=1 i zatim otvori bilo koja druga datoteka njen FH automatski dobija broj 1 čime je svaki ispis preko standardnog izlaza usmeren u nju Ovakav metod koristi COMMAND COM kada u komandnoj liniji naiđe na znak preusmeravanja > U ovom kontekstu se nazire i smisao datoteke sa FH=2. Ako se izlaz i preusmeri u neku datoteku i dalje nam je na raspolaganju kanal za ispis na ekran kako bi poruke o greškama bili vidljiva u toku rada programa

Mali nedostatak FH metoda će postati očigledan kada kažemo da DOS omogućava istovremeno korišćenje samo 8 brojeva Pošto je 5 već zauzeto ostaju slobodna samo tri veze Ovaj nedostatak se lako otklanja uvođenjem parametra FILES u CONFIG SYS datoteku na primer FILES=16

čime se opseg mogućih brojeva povećava na 11 (maksimalno do 99), ali zato druge korisnike vaših programa morate posebno upozoriti na ovu činjenicu na koju nemate uticaja Ipak programi koji rade sa više od tri istovremeno otvorena datoteke su veoma retki

Katalog MS DOS funkcija

Kod pregleda funkcija koristićemo sledeću notaciju i skraćenice

BROJ FUNKCIJE NAZIV FUNKCIJE
ORIGINALNI NAZIV FUNKCIJE

ULAZ Lista ulaznih parametara

IZLAZ Lista izlaznih vrednosti
GREŠKE spisak brojeva mogućih grešaka prema tablici ili šire objašnjenje

Uz prevedeni naziv funkcije ostavili smo i originalni zbog lakšeg praćenja strane literature koja je uglavnom na engleskom

Kod liste ulaznih parametara i izlaznih vrednosti koristićemo sledeće skraćenice
SEG (var) — SEGMENTNI DEO ADRESE NEKE VARIJABLE ILI STRUKTURE
OFS (var) — OFSETNI DEO ADRESE
kao i standardne naziva mikroprocesorskih registara.

Koriste se i već pomenuti nazivi
PSP — PROGRAM SEGMENT PREFIX
OFS (var) — OFSETNI DEO ADRESE

Kod liste ulaznih parametara i izlaznih vrednosti koristićemo sledeće skraćenice

Koriste se i već pomenuti nazivi
PSP — PROGRAM SEGMENT PREFIX
FCB — FILE CONTROL BLOCK
DTA — DISK TRANSFER AREA
FH — FILE HANDLE

Trudili smo se da gde god to prostor dozvoljava damo što celovitije primere Najveći broj primera predstavlja na žalost ipak samo male funkcionalne celine ali nadamo se ipak dovoljno korisne za objašnjenje korišćenja

Metod prijavljivanja grešaka nije jedinstven ali se ipak deli u samo dva grupa prema vremenu nastanka funkcija. Za novije funkcije kojima ćemo posvetiti više prostora koristi se CARRY zastavica Ako je po povratku u program CY=1 u redu funkcije je došlo do greške Broj greške će se nalaziti u registru AX Ako pored linije sa nazivom GREŠKE stoji samo jedan ili više brojeva onda funkcija koristi navedeni metod a brojevi imaju sledeće značenje

1 nevažeći broj funkcije
2 datoteka ne postoji
3 staza (PATH) ne postoji
4 suviše otvorenih datoteka nema više slobodnih FH
5 pristup nije odobren — najčešće zahtev za zapisivanje u datoteku sa atributom samo za čitanje (READ—ONLY)
6 upotrebljen nepostojeći FH
7 uništeni blokovi za kontrolu raspodele memorije
8 nedovoljno memorije da se zadovolji zahtev
9 pogrešna adresa memorijskog bloka — upotrebljena je adresa koja ne odgovara ni jednom bloku u sistemu odvajanje memorije
10 nevažeći blok okruženja (ENVIRONMENT)
11 nevažeći format
12 nevažeći kod za pristup
13 nevažeći podaci
14 —
15 nevažeći broj disk jedinice
16 pokušaj da se ukloni aktivni direktorij
17 nije isti uređaj i
18 nema više datoteka koje odgovaraju

Kod funkcija koje ne koriste ovaj metod biće dato posebno objašnjenje

U pregledu funkcije se nalaze sve koje se mogu bez rezerve koristiti u verzijama DOS-a od 2 00 do 3 20 Određen broj funkcija postoji samo u verzijama od 3 0 ali se danas teško možemo osloniti isključivo na njihovo korišćenje jer se DOS 2 10 (2 11) još uvek često upotrebljava Izostavili smo i funkcije kojima se DOS obogaćuje u prisustvu različitih hardverskih dodataka kao što su Microsoft Network EMS — EXTENDED MEMORY STANDARD i slično

Posebna napomena DOS se u velikom broju slučajeva oslanja na formate podataka koji se koriste u jeziku C Tako svi nazivi datoteka treba da završavaju bajtom 0 kao oznakom kraja brojevi od četiri bajta za koje se koriste po dva registra odgovaraju formatu LONG promenjivih (bajtovi poređani po težini od najnižeg ka najvišem) itd

00H PREKIDA RAD PROGRAMA

PROGRAM TERMINATE

ULAZ AH=00H
CS=SEG (PSP)
IZLAZ —
GREŠKE —

Jedan od manje preporučljivih načina završetka rada programa Ostatak ranijih verzija Kao preporučljiv metod vidi funkciju 4C

01H UPIS ZNAKA SA STANDARDNOG ULAZA SA EHOM

CHARACTER INPUT WITH ECHO

ULAZ AH=01
IZLAZ AL=UPISANI ZNAK
GREŠKE

27H ČITANJE PROIZVOLJNOG BROJA SLOGOVA

RANDOM BLOCK READ

ULAZ AH=27H
CX=BROJ SLOGOVA
DS=SEG (FCB)
DX=OFS (FCB)
IZLAZ AL=0 — OK
CX=BROJ PROČITANIH SLOGOVA
GREŠKE AL=1 — KRAJ DATOTEKE
AL=2 — PREKORAČENJE GRANICE SEGMENTA
AL=3 — PROČITAN SAMO DEO TRAŽENOG BLOKA NAIŠAO KRAJ DATOTEKE

Funkcija se razlikuje od svog jednostavnijeg ekvivalenta (21H) u dva elementa (1) može se čitati više od jednog sloga i (2) sledeće čitanje ide od pozicije gde je prethodno stalo DTA mora biti dovoljno velik da primi traženi broj slogova

28H ZAPISIVANJE PROIZVOLJNOG BROJA SLOGOVA

RANDOM BLOCK WRITE

ULAZ AH=28H
CX=BROJ SLOGOVA
DS=SEG (FCB)
DX=OFS(FCB)
IZLAZ AL=0 — OK
CX=BROJ PROČITANIH SLOGOVA
GREŠKE AL=1 — DISK PUN
AL=2 — PREKORAČENJE GRANICE SEGMENTA

Funkcija se razlikuje od svog jednostavnijeg ekvivalenta (21H) u dva elementa (1) može se čitati jednim pozivom zapisati više od jednog sloga (2) sledeće zapisivanje ide od pozicije gde je prethodno stalo i (3) ako se funkcija pozove sa CX=0 nema zapisivanja podataka već se datoteka produžuje ili skraćuje na veličinu određenu trenutnom pozicijom za zapisivanje

29H RAŠČLANJUJE NAZIV DATOTEKE

PARSE FILENAME

ULAZ AH=29H
AL BIT 3=1 MENJA PRODUŽETAK NAZIVA U FCB SAMO AKO JE ZADAT U NAZIVU DATOTEKE
=0 MENJA GA U SVAKOM SLUČAJU AKO NE POSTOJI U NAZIVU POPUNJAVA SA BLANKO
BIT 2 ISTO KAO BIT 3 ALI VAŽI ZA NAZIV DATOTEKE
BIT 1 ISTO KAO BIT 3 ALI VAŽI ZA BROJ DISK JEDINICE
BIT 0=1 IGNORIŠE RAZMAK ISPRED NAZIVA
=0 UZIMA I RAZMAK ISPRED NAZIVA
IZLAZ AL=0 — OK
AL=1 — NAZIV SADRŽI DŽOKER ZNAKE
ES=SEG (FCB)
DI=OFS (FCB)
GREŠKE AL=OFFH — OZNAKA DISKA NEVAŽEĆA

Funkcija treba da pomogne pri popunjavanju FCB kada je naziv datoteke i disk dat u uobičajenom korisničkom obliku D NAZIV PRO Put od ovog oblika do rastavljenih i delova naziva i broja disk jedinice često može da bude komplikovan za program uzimajući u obzir šta je korisnik sve u stanju da otkuca Ovom funkcijom se proces pojednostavljuje ali s obzirom da spada u red FCB funkcija sve se ređe koristi

2AH ČITA DATUM SA SISTEMSKOG KALENDARA

GET SYSTEM DATE

ULAZ. AH=2AH
IZLAZ CX=GODINA (1980—2099)
DH=MESEC
DL=DAN
AL=DAN U NEDELJI (0=NEDELJA)
GREŠKE —

22H ZAPISIVANJE SA DIREKTNIM PRISTUPOM
RANDOM WRITE

```
ULAZ    AH=22H
        DX=SEG (FCB)
        DX=OFS (FCB)
IZLAZ   AL=0 — OK
GREŠKE  AL=1 — DISK PUN
        AL=2 — PREKORAČENJE GRANICE SEGMENTA
```

Blok čija je dužina definisana nakon otvaranja datoteka u FCB prenosi se iz DTA na disk. Ne mora nužno proizvesti i fizičko zapisivanje podataka zbog internog baferisanja DOS-a. Nakon zapisivanja pozicija za sledeće zapisivanje ostaje na istom mestu. Program mora postaviti željeni redni broj sloga kao kod 21H.

23H VELIČINA DATOTEKE U SLOGOVIMA
GET FILE SIZE IN RECORDS

```
ULAZ    AH=23H
        DS=SEG (FCB)
        DX=OFS (FCB)
IZLAZ   AL=0
        FCB POLJE OD OFS 21H POPUNJENO
GREŠKE  AL=0FFH — DATOTEKA NE POSTOJI
```

FCB se koristi neotvoren (nije potrebna funkcija 0FH). Popunjava se nazivom datoteke i dužinom sloga koji služi kao jedinica.

24H POSTAVLJA REDNI BROJ SLOGA
SET RANDOM RECORD NUMBER

```
ULAZ    AH=24H
        DX=SEG (FCB)
        DX=OFS (FCB)
IZLAZ   MODIFIKOVANI PODACI U FCB
GREŠKE  —
```

Funkcija se koristi pri prelasku sa sekvencijalnog čitanja na direktan pristup.

25H POSTAVLJA INTERAPT VEKTOR
SET INTERRUPT VECTOR

```
ULAZ    AH=25H
        AL=BROJ INTERAPTA
        DX=SEG (NOVA INT RUTINA)
        DX=OFS (NOVA INT RUTINA)
IZLAZ   —
GREŠKE  —
```

Interapt vektori se nalaze u prvom kilobajtu RAM memorije i svaki zauzima 4 bajta (dve reči). Prva reč je ofset, druga segment adrese rutine koja će biti pozvana na odgovarajući interapt. Direktno menjanje vrednosti se ne preporučuje ako program treba da radi pod budućim multitasking verzijama DOS-a. Preporučljiva je praksa da se vrednost vektora koji se menja prethodno pročita (funkcija 35H) i sačuva pa da mu se pre izlaska iz programa vrati originalna vrednost. Ovo se ne odnosi na vektore za INT 22H, 23H i 24H kojima DOS vraća vrednosti koje je sačuvao u PSP.

Primer. Preuzimanje obrade CTRL-C interapta

```
CODE        SEGMENT
            MOV    AH 25H
            MOV    AL 23H
            MOV    DS SEG CTRLC-HANDLER
            MOV    DX OFFSET CTRLC-HANDLER
            INT    21H
CTRLC-HANDLER
```

Slika

Funkcija upisuje karakter sa standardnog ulaza koji može biti preusmeren redirekcijom. U tim slučajevima ne postoji način da se detektuje kraj datoteke (EOF). U normalnim situacijama upis ide sa tastature. Čeka se da korisnik pritisne taster (ako to već nije uradio), znak se istovremeno ispisuje na ekranu i ujedno se detektuje pritisak na CTRL-C što izaziva interapt 23H i prekid programa. Upis proširenih kodova zahteva dve poziva. Ako prvi poziv vrati kod 0 sledeći će vratiti prošireni kod pritisnutog tastera.

Primer. Čeka pritisak na taster ret

```
CODE SEGMENT
L1  MOV AH 1
    INT 21H
    CMP AL 0DH
    JNZ L1
```

02H ISPIS ZNAKA NA STANDARDNI IZLAZ
CHARACTER OUTPUT

```
ULAZ    AH=02H
        AL=KOD KARAKTERA
IZLAZ   —
GREŠKE  —
```

Funkcija ispisuje znak na standardni izlaz koji može biti i preusmeren redirekcijom. U ovim situacijama nemoguće je detektovati situaciju kada je disk pun. Ako u toku ispisa DOS detektuje CTRL-C taster izvodi se interapt 23 i program se prekida.

03H UPIS ZNAKA SA RS232 INTERFEJSA
AUXILIARY INPUT

```
ULAZ    AH=03H
IZLAZ   AL=UPISANI ZNAK
GREŠKE  —
```

Karakter se upisuje sa standardnog pomoćnog ulaza koji je inicijalno usmeren na prvi RS232 interfejs (COM1). Može se preusmeriti eksternom DOS naredbom MODE. Parametri komunikacije inicijalno su postavljeni na 2400 baud-a bez paritije sa jednim stop bitom i dužinom podatka od 8 bitova. U toku ispisa se detektuje i pritisak na CTRL-C i u tom slučaju izvodi interapt 23.

Mogućnosti upotrebe ove funkcije su veoma skromne. Komunikacija preko RS232 interfejsa zahteva stalnu proveru statusa komunikacije kao i siguran upis karaktera pre pojave sledećeg što ova funkcija nije u stanju da ispuni. Za sigurnu komunikaciju moraju se koristiti funkcije BIOS-a (INT 14) i čitanje karaktera posebno napisanom interapt rutinom.

04 ISPIS ZNAKA NA RS232 INTERFEJS
AUXILIARY OUTPUT

```
ULAZ    AH=04H
        AL=KOD ZNAKA
IZLAZ   —
GREŠKE  —
```

Karakter se ispisuje na standardni pomoćni izlaz koji je inicijalno usmeren na RS232 interfejs i može se preusmeriti eksternom DOS naredbom MODE. Detektuje se CTRL-C pritisak i izvodi interapt 23. Funkcija je najčešće u stanju da podrži samo ispis na štampač koji ima ovaj interfejs. Za punu dvosmernu komunikaciju u kombinaciji sa funkcijom 03H se gotovo i ne može koristiti.

05 ISPIS ZNAKA NA ŠTAMPAČ
PRINTER OUTPUT

```
ULAZ    AH=05H
        DL=KOD ZNAKA
IZLAZ   —
GREŠKE  —
```

Ako nije korišćeno preusmeravanje ispisa naredbom MODE, znak će biti poslat na prvi paralelni interfejs (PRN ili LPT1). U toku ispisa detektuje se CTRL-C taster.

Primer ispis stringa na štampač

```
         CODE      SEGMENT
                   MOV       BX OFFSET DUŽINA
                   MOV       CL BYTE PTR [BX]
                   XOR       CH,CH
         l1        INC       BX
                   MOV       DL BYTE PTR [BX]
                   MOV       AH 05H
                   INT       21H
                   LOOP      l1

         DATA      SEGMENT
         DUŽINA    DB        10
                   DB        'RAČUNARI'
                   DB        13 10
```

Slika 11

06H DIREKTAN UPIS/ISPIS SA KONZOLE
DIRECT CONSOLE I/O

ULAZ AH=06H
DL=00H — 0FEH — KOD ZNAKA — ZAHTEV ZA ISPIS
0FFH — ZAHTEV ZA UPIS
IZLAZ (ako je DL bio FFH)
Z=1 — TASTER NIJE PRITISNUT
Z=0 — TASTER JE PRITISNUT
AL=KOD TASTERA
GREŠKE —

Funkcija obezbeđuje gotovo direktnu vezu sa BIOS rutinama za čitanje tastature i ispis na ekran bez intervencije operativnog sistema. Ako je u toku ispisa/upisa pritisnut CTRL-C taster zahtev za prekidom se u potpunosti ignoriše. Upis/ispis mogu biti preusmereni sa svim prednostima i posledicama koje su data kod ranijih funkcija. Pri upisu sa tastature nema čekanja na pritisak na taster. ZERO flag služi kao indikacija da li je neki taster do trenutka poziva funkcije pritisnut. Za čitanje proširenih kodova potrebna su dva poziva kao kod 01H.

07H NEFILTRIRANI UPIS SA STANDARDNOG ULAZA BEZ EHOA
UNFILTERED CHARACTER INPUT WITHOUT ECHO

ULAZ AH=07H
IZLAZ AL=KOD ZNAKA
GREŠKE —

Funkcija je identična funkciji 01H, osim što nema ispisa upisanog znaka na ekran (standardni izlaz) i nema detekcije CTRL-C stanja.

08H UPIS ZNAKA SA STANDARDNOG ULAZA BEZ EHOA
CHARACTER INPUT WITHOUT ECHO

ULAZ AH=08H
IZLAZ AL=KOD ZNAKA
GREŠKE —

Funkcija je identična funkciji 01H osim što nema ispisa upisanog znaka na standardni izlaz. Detektuje se CTRL-C i izvodi interapt 23H. Funkcije 01H do 08H imaju mnogo dodirnih tačaka i praktično predstavljaju različite varijante sličnih operacija. Radi preglednosti sumiraćemo njihove karakteristike jednom tabilcom

FUNKCIJA	LOGIČKI UREĐAJ	SMER	EHO	CTRL-C	ČEKA ZNAK	REDIREKCIJA
01H	CON	U	DA	DA	DA	DA
02H	CON	I	—	DA	—	DA
03H	AUX (COM1)	U	NE	DA	DA	MODE NAREDBOM
04H	AUX (COM1)	I	—	DA	—	MODE NAREDBOM
05H	PRN (LPT1)	I	—	DA	—	MODE NAREDBOM
06H	CON	U/I	NE	NE	NE	DA
07H	CON	U	NE	NE	DA	DA
08H	CON	U	NE	DA	DA	DA

19H AKTIVNA DISK JEDINICA
GET DEFAULT DISK DRIVE

ULAZ AH=19H
IZLAZ AL=BROJ DISKA (0=A, 1=B ITD)
GREŠKE —

Funkcija u registru AL vraća broj u tom trenutku aktivne disk jedinice

1AH POSTAVLJA DTA
SET DTA ADDRES

ULAZ AH=1AH
DS=SEG (DTA)
DX=OFS (DTA)
IZLAZ —
GREŠKE —

Pre korišćenja funkcija za čitanje i zapisivanje koje koriste FCB metod obavezno je postaviti adresu područja u koje će biti preneseni podaci. Ako se ove funkcije ni jednom u toku programa ne pozove biće korišćena druga polovina PSP (ofset 80H). Svaki program je odgovoran da je DTA dovoljno velik za podatke koji se prenose. Nije dozvoljeno koristiti adrese na gornjoj granici segmenta kod kojih ofset usled ograničenosti na vrednost 65535 može preći na vrednost 0.

1BH INFORMACIJA O JEDINIČNOM ODVAJANJU PROSTORA NA AKTIVNOM DISKU
GET ALLOCATION INFO FOR DEFAULT DRIVE

ULAZ AH=1BH
IZLAZ AL=BROJ SEKTORA KOJI ČINE KLASTER
CX=VELIČINA FIZIČKOG SEKTORA DISKA (U BAJTOVIMA)
DX=BROJ KLASTERA NA DISKU
DS (BX)=0FFH — DVOSTRANI FLOPI, 8 SEKTORA/TRAGU
=0FEH — JEDNOSTRANI FLOPI 8 SEKTORA /TRAGU
=0FDH — DVOSTRANI FLOPI 9 SEKTORA /TRAGU
=0FCH — JEDNOSTRANI FLOPI 9 SEKTORA /TRAGU
=0F9H — DVOSTRANI FLOPI, 15 SEKTORA /TRAGU (IBM PC-AT)
=0F8H — TVRDI DISK
GREŠKE —

Ukupan kapacitet medijuma se dobija množenjem AL*CX*DX. Klaster predstavlja minimalnu jedinicu odvajanja prostora na disku

1CH INFORMACIJA O JEDINIČNOM ODVAJANJU PROSTORA NA SPECIFICIRANOM DISKU
GET ALLOCATION INFO FOR SPECIFIED DRIVE

ULAZ AH=1CH
DL=BROJ DISKA (0=AKTIVAN, 1=A 2=B ITD)
OSTATAK PODATAKA IDENTIČAN SA 1BH

21H ČITANJE SA DIREKTNIM PRISTUPOM
RANDOM READ

ULAZ AH=21H
DS=SEG (FCB)
DX=OFS (FCB)
IZLAZ AL=0 — OK
GREŠKE AL=1 — KRAJ DATOTEKE
AL=2 — PREKORAČENJE GRANICE SEGMENTA
AL=3 — PROČITAN SAMO DEO TRAŽENOG BLOKA, NAIŠAO KRAJ DATOTEKE

Blok čija je dužina definisana nakon otvaranja datoteke u FCB prenosi se sa diska u DTA koji prethodno mora biti postavljen funkcijom 1AH. Nakon čitanja pozicija za sledeće čitanje ostaje na istom mestu. Program mora postaviti željeni redni broj sloga (četiri bajta od ofseta 21H u FCB)

```
          MOV     AH, 1AH
          INT     21H  postavlja DTA
          MOV     DX  OFFSET FCB 1
          MOV     AH  14H
          INT     21H  sekvencijalno čitanje

DATA      SEGMENT
FCB 1     DB      0
          DB      DATA
          DB      G
          DB      25 DUP (0)
DTA       DB      128 DUP (I)
```

Slika 14

15H SEKVENCIJALNO ZAPISIVANJE
SEQUENTIAL WRITE

```
ULAZ    AH=15H
        DS= SEG (FCB)
        DX= OFS (FCB)
IZLAZ   AL=0 — OK
GREŠKE  AL=1 — DISK PUN
        AL=2 — PREKORAČENJE GRANICE SEGMENTA
```

Važe iste napomena kao i za 14 H

16H KREIRA (PONIŠTAVA) DATOTEKU
CREATE OR TRUNCATE FILE

```
ULAZ    AH=16H
        DX= SEG (FCB)
        DX=OFS (FCB)
IZLAZ   AL=0
GREŠKE  AL=0FFH — NEMA MESTA U DIREKTORIJU
```

Kreiranje nove datoteka automatski poništava sveki sadržaj eventualno postojeće sa istim nazivom — bez upozorenja. Nakon funkcije datoteka ostaje otvorena za upis/ispis. Ako se upotrebi prošireni FCB datoteci se u toku kreiranja može dodeliti neka od kombinacija atributa

17H MENJA NAZIV DATOTEKE
RENAME FILE

```
ULAZ    AH=17H
        DX= SEG (FCB)
        DX=OFS (FCB)
IZLAZ   AL=0
GREŠKE  AL=0FFH — ORIGINALNI NAZIV NE POSTOJI ILI NOVI NAZIV VEĆ POSTOJI
```

Struktura FCB za ovu priliku ime posaban oblik (vidi primer). Nazivi mogu da sadrže i džoker znaka

Primer — promena svih naziva koji počinju sa DAT tako da počinju sa TST

```
CODE      SEGMENT
          MOV     AH  17H
          MOV     DX  OFFSET FCB SPEC
          INT     21H
          CMP     AL  OFFH
          JE      ERROR

DATA      SEGMENT
FCB-SPEC  DB      0
          DB      DAT?????'
          DB      ???
          DB      5 DUP (0)
          DB      TST?????
          DB      ???
          DB      15 DUP (0)
```

Slika 15

09 ISPIS STRINGA NA STANDARDNI IZLAZ
OUTPUT CHARACTER STRING

```
ULAZ    AH=09H
        DS=SEG (STRING)
        DX=OFS (STRING)
IZLAZ   —
GREŠKE  —
```

Funkcija je izvedena na osnovu funkcije 02H Niz karaktera koji se ispisuje mora biti obeležen znakom $ na kraju ($ se ne ispisuje)

Primer ispis poruke

```
CODE      SEGMENT
          MOV     DX OFFSET STRING
          MOV     AH 09H
          INT     21H

DATA      SEGMENT
STRING    DB RAČUNARI 13 10 $
```

Slika 12

(Napomena U primeru se podrazumeva da je DS već ranije dobio vrednost Vidi šablon programa u uvodu)

0AH UPIS STRINGA SA EDITOVANJEM
BUFFERED INPUT

```
ULAZ    AH=0AH
        DS=SEG (BUFFER)
        DX=OFS (BUFFER)
IZLAZ   STRING I NJEGOVA DUŽINA U BAFERU
GREŠKE  —
```

Funkcija je izvedena korišćenjem funkcije 01H, pa važe iste napomene za redirekciju CTRL-C itd Bafer u koji će biti upisan string sa tastature ima posebnu strukturu prvi bajt treba da sadrži broj znakova koje bafer može da primi, čime se upis može ograničiti na željeni broj znakova (najviše 255) Drugi bajt popunjava DOS na izlazu i sadrži broj znakova koji je stvarno upisan (uključujući i kod 13 — RET) Za njime sledi upisan string

Ovo je standardna funkcija za upis stringa koju koristi i COMMAND COM tako da poseduje sve poznate editna mogućnosti — brisanje poslednjeg znaka BACKSPACE tasterom ignorisanje upisa ESC tasterom ponavljanje prethodnog upisa tasterima F1 F2 i F3 itd

Primer Odgovor na pitanje

```
CODE      SEGMENT
          MOV     DX OFFSET PITANJE
          MOV     AH 09H
          INT     21H
          MOV     DX OFFSET ODGOVOR
          MOV     AH 0AH
          INT     21H

DATA      SEGMENT
PITANJE   DB      NAZIV DATOTEKE $
ODGOVOR   DB      66 dozvoljena dužina
          DB      0 dužina upisanog stringa
          DB      66 DUP (0) upisani string
```

Slika 13

0BH STATUS ULAZA
GET INPUT STATUS

```
ULAZ    AH=0BH
IZLAZ   AL=00H — ZNAK SPREMAN ZA ČITANJE
        AL=0FFH — NEMA SPREMNOG ZNAKA
```

Funkcija proverava da li postoji znak na standardnom ulazu koji nije pročitan. Iako se ulaz može preusmeriti funkcija se najčešće koristi za proveru da li u baferu tastature postoji nepročitan taster. U slučaju detekcije CTRL C izvodi se interapt 23H

0CH RESETUJE ULAZNI BAFER I UPISUJE ZNAK ILI STRING
RESET INPUT BUFFER AND THEN INPUT

```
ULAZ    AH=0CH
        AL=BROJ FUNKCIJE ZA UPIS
        (01H 06H 07H, 08H 0AH)
IZLAZ   vidi tekst
GREŠKE  —
```

Funkcija omogućava poziv bilo koja funkcije za upis znaka ili stringa sa standardnog ulaza koja je do sada pomenuta s tim što se prethodno poništavaju svi znaci koji eventualno postoje u baferu a nisu pročitani. Ulazni i izlazni parametri su isti kao kod funkcija koja se poziva (npr 0AH zahteve OS DX itd) kao i sve ostale karakteristike pojedinih funkcija

0DH RESETUJE DISK JEDINICE
DISK RESET

```
ULAZ    AH=0DH
IZLAZ   —
GREŠKE  —
```

Naziv ove funkcije nije do kraja adekvatan njenom dejstvu. Svi podaci upućeni na disk, zbog optimizacije rada čitavog sistema privremeno se smeštaju u interne bafere i samo povremeno u određenim porcijama fizički zapisuju na disk. Ova funkcija omogućuje da se u bilo kom trenutku podaci u baferu koji još nisu zapisani na disk prenesu na magnetni medijum.

Funkcija se retko koristi popotovo što sama po sebi ne obezbeđuje kasniju dostupnost podataka. Ako se upotrebi i podaci stvarno zapišu na disk a ne izvede propisna procedura zatvaranja datoteke nastale promena u veličini datoteke neće biti unata u direktorij diska što podatke čini nedostupnim

0EH ODREĐUJE AKTIVNU DISK JEDINICU
SET DEFAULT DISK DRIVE

```
ULAZ    AH=0EH
        AL=BROJ DISK JEDINICE (0=A 1=B ITD )
IZLAZ   AL=BROJ LOGIČKIH DISK JEDINICA U SISTEMU
GREŠKE  —
```

Broj disk jedinice je njen logički broj u sistemu. DOS 2 10 je omogućavao korišćenja do 63 logičke jedinice (računajući RAM diskove disketne jedinice, tvrda diskove itd) ali je kasnija od verzije 3 0 broj smanjen na 26 kako bi se za naziva koristila samo slova abecede (A—Z). Interesantno je da proglašavanja nepostojeće disk jedinice za aktivnu ne proizvodi nikakvu grešku već jednostavno nema efekta. Za računare sa samo jednom diskom jedinicom se smatra da poseduju dve logička jedinice (A i B)

0FH OTVARA DATOTEKU
OPEN FILE

```
ULAZ    AH=0FH
        DS=SEG (FCB)
        DX=OFS (FCB)
IZLAZ   AL=0 FCB POPUNJEN
GREŠKE  AH=0FFH — DATOTEKA NE POSTOJI
```

Ova funkcija spada u grupu ostataka ranijih verzija DOS-a. U FCB treba popuniti naziv datoteke i broj disk jedinice (0=aktivna 1=A, 2=B itd). Nakon uspešnog otvaranja datoteke DOS popunjava ostatak FCB prema podacima iz direktorija diska i postavlja dužinu bloka na vrednost 128

10H ZATVARA DATOTEKU
CLOSE FILE

```
ULAZ    AH=10H
        DS=SEG (FCB)
        DX=OFS (FCB)
IZLAZ   AL=0
GREŠKE  AL=0FFH   DATOTEKA NE POSTOJI U DIREKTORIJU
```

Adresa FCB se mora odnositi na isti FCB kojim je otvarena

11H NALAZI PRVU ODGOVARAJUĆU DATOTEKU
SEARCH FOR FIRST MATCH

```
ULAZ.   AH=11H
        DS= SEG (FCB)
        DX=OFS (FCB)
IZLAZ.  AL=0 DTA POPUNJEN KAO STANDARDNI ILI
             PROŠIRENI FCB
GREŠKE  AL=0FFH — NI JEDNA DATOTEKA NE ODGOVARA ZADATOM NAZIVU
```

Funkcija služi za pronalaženja podataka o datotekama na disku. Pre poziva je obavezno postaviti DTA. Naziv datoteka u FCB može sadržati u džoker znake (* ?). Prošireni FCB se koristi kada se traže nazivi i podaci za datoteke obeležena nekim posebnim atributom (READ-ONLY SYSTEM itd). Ukoliko je ulaz u funkciju prošireni FCB izlaz u DTA je takođe takav

12H NALAZI SLEDEĆU DATOTEKU
SEARCH FOR NEXT MATCH

```
ULAZ    AH=12H
        DS=SEG (FCB)
        DX=OFS (FCB)
IZLAZ   AL=0 DTA POPUNJEN PODACIMA ZA SLEDEĆU DATOTEKU
GREŠKE  AL=0FFH — NEMA VIŠE DATOTEKA ČIJI NAZIV ODGOVARA
```

Funkcija pretpostavlja da je prethodno izvedena 11H DOS nastavlja sa popunjavanjem DTA na opisani način sve dok ni jedan naziv (ili atribut kod proširenog FCB) više ne odgovara zadatom kriterijumu

(Za preporučljiv metod čitanja sadržaja direktorija vidi nove funkcija 4EH i 4FH)

13H BRIŠE DATOTEKU
DELETE FILE

```
ULAZ    AH=13H
        DS=SEG (FCB)
        DX=OFS (FCB)
IZLAZ   AL=0
GREŠKE  AL=0FFH — NIJEDNA DATOTEKA NIJE OBRISANA
```

U FCB se popunjava samo naziv datoteka koja se briše. Naziv može sadržati i džoker znake pri čemu će biti obrisane sve datoteka čiji nazivi odgovaraju. Ako nijedna datoteka nije obrisana ili ni jedan naziv ne odgovara ili su datoteke koje odgovaraju označene atributom READ-ONLY vraće u AL kod 0FFH

14H SEKVENCIJALNO ČITANJE
SEQUENTIAL READ

```
ULAZ    AH=14H
        DS=SEG (FCB)
        DX=OFS (FCB)
IZLAZ   AL=0 — OK
        AL=1 — KRAJ DATOTEKE
        AL=2 — PREKORAČENJE GRANICE SEGMENTA
        AL=3 — PROČITAN SAMO DEO TRAŽENOG BLOKA  NAIŠAO KRAJ DATOTEKE
GREŠKE  —
```

FCB mora biti isti onaj koji je prethodno korišćen za funkciju OPEN. DTA područje se ne sme nalaziti blizu granice segmenta tako da povećanja ofsetne vrednosti pri prenošenju pojedinačnih bajtova sa diska može zbog ograničenosti registra na 64K ponovo praći na nulu. Svaki sledeći poziv čita od mesta gde je prethodni stao

Primer: Otvaranje i čitanje prvih 128 bajtova datoteke

```
CODE    SEGMENT
        MOV     AH 0FH
        MOV     DX OFFSET FCB_1
        INT     21H otvara datoteku
        CMP     AL 0FFH
        JE      ERROR
        MOV     DX OFFSET DTA
```

Mali oglasi

Ako ne možete da podnesete da drugi nemaju ono što vi imate, objavite svoj mali oglas u „Računarima"

Ako ne možete da podnesete da drugi imaju ono što vi nemate, javite se na neki od malih oglasa u „Računarima'

Prva stvar koju treba da uradite jeste da se odlučite da li želite običan ili uokviren mali oglas.

Obični oglasi

Cena običnog malog oglasa do dvadeset reči je 1800 dinara Svaka naredna reč košta još 150 dinara. Veznici, predlozi prilozi, zamenice, brojevi i ostale „male" reči se računaju u cenu Adresa oglašivača se ne računa u cenu Tri važna ograničenja. mali oglas ne može biti duži od 50 reči mali oglas ne može biti štampan velikim slovima i uz mali oglas se ne mogu objavljivati crteži i fotografije Mali oglasi koji se ne uklapaju u ove okvire pripadaju kategoriji uokvirenih malih oglasa

Uokvireni mali oglasi

Cena uokvirenog malog oglasa je 2000 dinara i po visinskom centimetru u stupcu širine 9,5 cicera ako oglas nije viši od pet centimetara i 3000 dinara po visinskom centimetru ako je mali oglas visok između pet i deset centimetara U sklopu uokvirenog malog oglasa mogu se objavljivati fotografije i crteži i mogu se birati veličine i tipovi slova (belo poluceno, kurziv) Fotografije i crteži se plaćaju prema prostoru kao da se radi o tekstu Jedno važno ograničenje uokvireni oglasi preko 10 centimetara ne spadaju u kategoriju malih oglasa Za njih važe pune komercijalne cene i oni se ugovaraju sa oglasnim odeljenjem BIGZ-a

Pripreme malih oglasa

Poželjno je da mali oglas počinje sa Prodajem, Kupujem, Držim časove Menjam i sličnim što ukazuje na njegovu sadržinu Adresa oglašivača se kuca u produžetku teksta malog oglasa a ne odvojeno Ova ograničenja razume se ne važe za uokvirene male oglase

Uz mali oglas treba navesti njegovu vrstu (običan, uokviren) i kategoriju u kojoj će biti objavljen „spektrum", „komodor", „hardver", „literatura" [illegible])

Prijem i plaćanje malih oglasa

Mali oglas treba dostaviti na adresu redakcije „Računari' — BIGZ (za male oglase), Bulevar vojvode Mišića 17, 11000 Beograd najkasnije do petog u mesecu Sv oglasi koji do ovog roka pristignu u redakciju poštom, lično i, uz određena ograničenja, telefonom, biće uvršćeni u sledeći broj

Mali oglasi se, po pravilu, plaćaju unapred bankovnom uplatnicom na račun 60802-603-23264 BIGZ, Bulevar vojvode Mišića 17, 11000 Beograd sa obaveznom naznakom „Računari", mali oglasi Kopiju uplatnice treba, obavezno, dostaviti zajedno sa tekstom malog oglasa

Prijem malih oglasa telefonom

Mali oglas se, u opravdanim slučajevima, može redakciji dostaviti i telefonom (011/650-161), ali samo pod uslovom da nije duži od dvadeset reči ili da se naručuje ponavljenje oglasa iz prethodnog broja „Računara' U tom slučaju oglas se može platiti i naknadno, uplatnicom koju će oglašavaču uputiti redakcija

Važno upozorenje

Sadržaj malog oglasa predstavlja diskreciono pravo oglašivača pod uslovom da ne dolazi u koliziju sa zakonom i da ne vređa javni moral U tom smislu u „Računarima" se ne može objaviti mali oglas u kome se naglašava ili veliča pornografska, militaristička ili slična društvenoneprihvatljiva osobine pojedinih igara

Spektrum

Spektrumovci!!! Ako vas interesuju najnovija softwerska dostignuća obratite se dema softu Snimanje je direktno iz Spektruma Tražite besplatan katalog. Silbaški Dejan Titogradska 8/1 21000 Novi Sad tel 021/369-861

SPEKTRUMOVCI PAŽNJA!!!
Futuresoft vam i ovaj mesec nudi puno novih i najnovijih programa za vaš ZX SPECTRUM Izabrati možete između 63 komplete igara i 6 kompleta uslužnih programa a pored toga i veliki broj specijalnih kompleta (šah sax simulacije) Futuresoft vam i nabavlja programe Cena kasete + poštarine + programa = 3000 dinara Naručite programe i vi budite jedan od onih koji mogu da kažu sa ponosom Da naručio sam programe kod Futuresofta na adresi POLJANSKI NASIP 30 61000 LJUBLJANA ili na telefonu 061-311-831 od 15 do 22 sata svaki dan.

Prodajem ZX Spectrum+2 (128 K) Tel 024/28-488 Vidicki Jovan 24000 Subotica Trg L Nešića 10/4

SPECTRUM RAINBOW SOFTWARE
Vam nudi preko 70 copy programa + superturbo za 1000 din Najkompletnije ponude od 2500 programa Mihajlovski Krčo Moše Pijade 12P 91300 Kumanovo tel 090/123-800

Kupujem ZX Spektrum 48K može i sa interfejsom i sa Džojstikom Obavezno navesti cenu Momir Šarić, 78428 Nožičko

Izaberite 18 programa + kaseta C90 = 2100 din Topgun Arcanoid Hypball Feud Sigma7 Bazookabill AgenX M Madnes Dandare Paperboy Lightforce, Uridium Elite N O M A D Dombjack 12 Tarzan Nosferatu Terracresta Xevious Scoobydoo Cobra TTRacer 1942 Greenbert Njinja BMX Samanthe F Šifal Vlado JNA 54 26210 Kovačice tel 013/781-318 Maroš (13—18h)

Spektrumavci Najnovije igre u turbu prodajem Nemesis Vehimata, World Games + 15 igara + kaseta = 3000 dinara Fećak Dinko Drvarska 36 42000 Varaždin

COMET SOFTWARE
SPEKTRUMOVCI!!!
Za one koji se više ne igraju već ozbiljne koriste svog ljubimca — NAMENSKI PROGRAMI svih vrsta — PROGRAMSKA UPUTSTVA bez kojih se ne može
NAJNOVIJE
— LASER GENIUS — Oasis
— LASER BASIC — Oasis
— MACHINE LIGHTNING — Oasis
— THE WRITER — Softechnics
— LAST WORD — Mymidonsoft
— OFFICE MASTER — Gemini
GARANCIJA ZA SVE VRSTE USLUGA
MILOVANOVIĆ LJUBIŠA Petra Lekovića 57, 11030 BEOGRAD tel 011/556-097 posle 17h 037046

Hawk-Soft vam nudi najbolje programe za vaš Spectrum Snimamo direktno iz kompjutera Naručite besplatan katalog na adresu Hawk-Soft Zdenački zavoj 18 41040 Zagreb ili Hawk-Soft Zdenački zavoj 17 41040 Zagreb

Spektrumovci Najnovije igre u turbu prodajem Nemesis Vehimata World Games + 15 igara + kaseta = 3000 dinara Fećek Dinko Drvarska 36 42000 Varaždin

Prodajem povoljno štampač TIMEX za ZX Spectrum i odgovarajuće rola beleg papira (na srvi termički) za njega širine 11 cm Dragan Sundać 51000 Rijeka, Žrt. fašizma 49 tel 061/24-673

Cobra Soft — najnoviji programi samo 100 din Saboteur 2 Nemesis of Warlock Indoor Sports Dracula Eidolon, Academy Head Over Heels Cobra Soft, Slavka Rodića 5A 23000 Zrenjanin tel. 023/64-223

Spektrumovci! Komplet od 12 najboljih igara za Spektrum možete dobiti za samo 1000 dinara Zajedno sa kasetom i poštarinom 2150 dinara. Dejan Stanković Nas Sretena Dudića gal 1/15 14000 Valjevo tel 014/36-540

Šah! Šah! Šah! Ljubitelji šaha! Proverite svoje sposobnosti Igrajte protiv najvećih šahista Cena kompleta (17 programa) je 1000 dinara Isporuka odmah. Dejan Stanković Nas Sretena Dudića gal 1/15 14000 Valjevo tel 014/36-540

Komodor

Super Soccer Enduro Racer Saboteur 2 i još 200 odabranih igara (upravo prispelih iz Engleske) Uz brzu i kvalitetnu uslugu. Katalog i cene idealne Vranić Dejan Njegoševa bb 19300 Negotin. 081/52-856 (od 13—18h.)

Super komplet — Spektrum komplet od 240 programa 2 000 din na vašim ili 4 800 na 4 moje kasete Spisak besplatan Savinovski S 43400 Virovitica GA jeva 4 tel 046/724 776

Spektrumovci! Mursa soft vam nudi uz nisku cenu veliki izbor najnovijih hitova. Snimanje iz Spektruma katalog besplatan Pruški Željko Bosanska 2 54000 Osijek

Spektrum bez kabla za napajanja Uređaj za vrlo jednostavnu ugradnju, sa „sinapsom" zajedno u kutiji Obaveštenja na tel 031/891-501 Jovanović Boban ili 031/891-684 Mićović Željko od 15—19 časova

YU PIRAT SOFT zbir vrhunskih piratskih dostignuća poziva vas na jedrenje po moru najnovijih engleskih hitova Isporuka brza kvalitet zagarantovan Yu Pirat Soft Kralja Milutina 5A 18000 Niš. 018/713-121

INTERES te opominje Nemaš vremena na prateki Uzmi kvalitet, a drugima ostavi kvantitet Pusti tradiciju i javi se Interes Pionirska 2b Sarajevo 071/37--256

Komodor 64/128 Vrhunski uslužni program za disk i za kasetu Uputstvo za korišćenje disk igra Miroslav Čakarevski Beograd Radoja Domanovića 28 011/417-371

Commodore 16 116 +4 najveći izbor najnovijih igara i uslužnih programa. Cene niske. Usluge kvalitetne. Uz svaku narudžbu poklanjam Turbo. Nazovite i uverite se. Dragan Ljubisavljević 3 oktobar 302/6 19210 Bor

Komodor 64 izaberite sami svoj komplet iz ostalih oglasa. Kataloge ne šaljem, jer posedujem sva programe. Miroslav Čakarevski Radoja Domanovića 26, Beograd 011/417-371

Spektrumovci! Najnoviji programi po 80 din. Kompleti su još jeftiniji. Izvanredan snimak. Prodajem nov Kemodor 64 sa kasetofonom. Tošev Mladan, M. Tita b/b. 37220 Brus

Commodore 64 Wizard and Warrior Atlantis Soldier II—III Turbo Esprite + Revenge II Enduro Racer Gunstar Deluxe Marble Madness + Ket Trilogy I—III Superstar Ice Hockey 30 programa + kaseta + ptt = 2500 din. Videnović Stanica, AVNOJ C-2/II-10 18000 Zaječar tel 019/28-600 (posle 15h)

Taliing Soft. Najbolji najjeftiniji, najnoviji programi za Komodor 64. Katalog besplatan. Usluge kvalitetne. Cena pojedinačnog programa iznosi samo 40 din. Požurite čekamo vas. Drašvić Krunoslav Prvomajska 72 22400 Ruma tel 022/421-360

Prodajem Kemodor 64 Disk 1541 Štampač Komodor MPS802+5 hiljada programa. Sve informacije na adresu. Adi Tinjić Senjak E4 75000 Tuzla tel 075/235-866

Kupujem sve programe (osim uslužnih) za C-64 od najstarijih do najnovijih. Dogovor po prijemu kataloga sa cenom svih programa. Bogojevac Stevan, Panonska 4D/3 25230 Kula

Komodor 64 Sve programe koje vidite u oglasima i još nešto možete kod nas dobiti po znatno nižim cenama. Katalog besplatan. Nikolić Dejan, Gligorije Dikića 52, Kruševac 037/33-753

Commodore 64 50 igara + kaseta=3000 din. 80 korisničkih programa + kaseta=3000 dinara Sašo Bezlaj Kraška Cesta 32/A 66215 Divača

C16 +4 prodajem najnovije programe po najnižim cenama. Proverite Jekšić Dejan Omera Maslića 20 71000 Sarajevo 071/617-283

C S H
COMODORE SOFT House od ovog meseca nabavlja samo za vas najnovije programe po najnižim cenama! Komplet od 30 programa (Feud Terminator Top Gun Eagles Nest ...) + kaseta = 1700 din. Pojedinačni programi 50 din. Zovite 011/541-862 Goran ili 011/512-584 Srđan. Mnogo zabave uz vaše nove programe!

Komodorovci pradajemo najnovije programe snimane direktno sa računara 30 programa + kaseta = 2500 din. Tražite besplatan katalog Klub M and M Milan Abrahamsberg Ljubljanska 13 61310 Ribnica. tel 061/861 161

Jedinstvena prilika. Prodajem 30 najnovijih programa po veoma povoljnoj ceni od 1500 din. Svi naručioci učestvuju u nagradnoj igri. Glavna nagrada Džojstik Kwikšot 2. Stanarević Đorđe Štrosmajerova 13 78000 Banjaluka

Prodajem igre i korisničke programe pojedinačno. Cena od 30—40 din. Katalog besplatan. Sabo Aleksander Redovan Kosarsko 35 21217 Bačko Gradište 021/806-001

Komplet 14 The jet (1—3) Enduro race, Superstar Iceknckey 2/2 Hockey 8/8 100 Millione Space Shuttle Toad Force Pakos Petty 2 Drum Time Gram Paina (1—3) Ring Side Boxing Turbo Esprit, Deceptor Soldier Nebuccah, Titans Brutallo Amaurote Space Hunter Strike. Premier Leagure Gun Star Center Fold Ket (1—3) 30 program = 1500 dinara Stanarević Đorđe Štrosmajerova 13 78000 Banjaluka tel 078/32-062 i 30-540

COMMODORE 64/128 Naručite profesionalna prevode. Svi prevodi su formata A4 i tvrdo ukoričeni Graph 64 Stat 64 po 700 d Trikovi po 1000 d Kontemat, Practical Multiplan po 1100 d Simons Basic po 1300 d Flight Simulation II Super Base po 1500 d Giga-Cad dBASEII Viši kurs mašinskog programiranja po 200 d C-64 uputstvo po 2500 d Minja Višnjić Jurija Gagarina 141/83 11070 Novi Beograd tel 011/157-758

Komodore Boy Software najnoviji programi po ceni od 40 din. Katalog besplaten. Mijatov Miljan Bul V Vlahovića 33/23 23000 Zrenjanin tel 023/68-458

Prodajem najnovije programe za C-64 koji nam stižu tjedno iz Holandije Nemačke i Engleske. Besplatan spisak. Zagreb Cracking Service — Dzren Đukić, 41020 Zagreb Čalugevićeva 5 tel 041/686-004

NEW NOW SOFT
Commodore 64 sve na jednom mestu. Sve što želite da vidite u svojoj kolekciji možete potražiti kod nas. Najveći hitovi maja i juna već vas čekaju Aufwied Monty Mario Bros Cyborg+ Krackout 3 i do izlaska broja još mnogo najnovijih programa. Dušan Adamović Titogradska 6 21000 Novi Sad tel 021/366-205

QCS Nabavite najnovije programe kod nas kao što su Turbo Esprit Enduro Racer Deuntlet 2 Army Moves 1,2 Profesionalna usluga i niske cene. Katalog. Cvijić Lazar Dragomira Gajića 89/19 37000 Kruševac tel 037/28-553

CBM 64 Miracle Soft objavljuje rasprodaju oko 130 pop programa na 3 kasete samo 3000 din. To jest 27 din po programu. Garancija 6 meseci stari kupci popust 50 posto. Robert Ranković Balilo 42 21205 Sremski Karlovci Tel: 021/861-909

Pojedinačno najnoviji programi za C-64 kasetni (Turbo Esprit Enduro Racer) 50 din. Disketni 300 din po strani diskete (Video Tril Highland) Petrović, Sarajačka 44 11000 Beograd tel 011/650-509

Za Kamodor 64 prodajem najnovije programe po niskim cenama Daniel Damjanović Musaleš bb 51440 Poreč

C-16/C-16+4 27 programa 3000 din Matrix Vox Autobahn Rocuman Speedung Kana Conwoy Goldrush Strippo ver Hewtik Spectipede Tutifruti Zodiac Wall Hyried Spacemission Zylon Light Force Swarhestiny Starwars 2 Speedboat Falcon Bandits Ó Profy Destruction Legionare Moonbyggy Laser Zone. Čobanov Branislav Petra Drapšina 53/1 21480 Srbobran tel 021/730-364 od 8—12 i 16—22 h

C-64 Superhitovi 34 programa 1200 din Adrianmole 1—3 Deuntlet 2(1—4) Dreatgscape Top Gun Armymers 1—2 Highland 1—3 Samyrai 1—3 Turbo EspritIII Beach Heade 3(pravi) Thenatos Nemesis Vainquere ub40 Duett Gunstar Psycatle Ufo Haktik, Fireball Castles Revelation Bubble Madness Gallery Čobanov Branislav P Drapšina 53/1 21480 Srbobran tel 021/730-364 od 8—12 i 16—24 h

Commodore 64! Turbotejp sa kasetom i vašim imenom 1300 din. Šaljite spiskove programa koje želite. Tražite katalog Top Gun Nosferatu ... Miljko Teo Rudarska 19 88000 Mostar tel 088/37-093

SHIFT-SOFT UVEK PRVI! Za letnji raspust nudimo 5 **kompleta** puna hitova! Sve što nuda ostali zajedno na jednom mestu!
KOMPLET 17 TURBO ESPRIT (hit sa Spectruma!) ANIMATED STRIP POKER FRANKIE GOES TO HOLLYWOOD 2 DRUG RACE ELIMINATOR 1 i 2 SPACE SHUTTLE SIMULATOR HEAD OVER HEELS BLACK NIGHT CIRCOPOLY CHAMPIONSHIP FOOTBALL SOLDIER SAMURAY THE JET (program sa diska!), DROWING PAINS OF ANDRIAN MOLE 2 EAGLE HUNT METROCROSS WARLOK TRAINER KET TRIOLOGY 3 MEGA SOUNDPACKER RINGSIDE BOXING RACE 5 DEVIL TOLLY YOYO DAWN PATROL, TULIP NOIR DRUM TIME STIF FLIP, KANDAR HOWERCRAFT NEBUCCA TOAD FORCE TRAPSTER A HEAVY THINK WOBBLY SHARKS LIEUTNANT LAWN ENCLAWE MAGIC CIRCLE 87 SPECTRAL SNEER
KOMPLET 18 PERISCOPE UP! BRUTAL BOULDER DASH BUGGSY 2 THE EQUILAZER TREASURE ISLAND (hit sa C16/plus 4!) EAGLES TEMPLE OF TERROR (sa diska) THE JET 2 FABSO M MIXABLE SOLDIER 2 ENDURO RACER (hit sa automata) BLIND VISION KET TRIOLOGY 1 ANDRIAN MOLE 3, MAD MONKEY REVENGE 2 POLLUTER CALIF GOLDRUSH PSSST! CASTLES OVER SEAS GUN STAR DRUG RACE ELIMINATOR 3 AUSTRALOPITICUS ROCK MONITOR III MILK RACE WASTED YEARS VAINQUEUR THAHATOS TRAINER FADE TO BLACK, SUPERSTAR 1, SPACE HUNTER RAGE HARD PREMIER LEAGUE BUDDY BUBBLE ANTLANTIS, SPACE CALLS
KOMPLET 19 DARK SCEPTRE (hit sa Spectruma) A KIND OF MAGIC, TITANS KET TRIOLOGY 2 MICROOLYMPICS TURBO ESPRIT II XENON RANGER CYLU MISSION II THEY STOLE A MILLION 2, DECEPTOR TIME TRAVELER FROG JUMPER S O I J GRIBBLY 2 DEMO DESIGNER MADNESS TRAINER ARMY'S 1st, VOID RUNNER TDC STRIKES THE JET 3 DRUG RACE ELIMINATOR 4 SOLDIER 3 GUN STAR DELUX WOOLOOGIE PIED PIPER AMAUROTE SUPERSTAR 2 BAKED BEANS ANDRIAN MOLE 1 BURGER RIOT STRIKE ZENITH GUN RUNNER 3D PACOS PETE II LETHAL PLAYBOY PREVIEW CENTERFOLD II
Komplet + kaseta + PTT = 3000 dinara. dinara. Dva kompleta za 5500 a sva tri za 8000 dinara! Za stare mušterije popusti! S H I F T — S O F T !
Vasović Nenad Dubravačka 19 11080 Zemun 011/210-684
Kremenović Goran Dušana Vukasovića 74 11070 Novi Beograd 011/172-234

Mastera Adi Software vam nudi za letnji raspust najbolje programe u kompletima (25 prog =1000 din + k+p) Pojedinačno 50 din. Velike povoljnosti. Paklen. Besplatan katalog. 51514 Dobrinj otok Krk

DOSC Disk programi Ice Hockey Sidney Affair Samurai Love Quest, Deltaman ... 1 program diskete i ptt 1800 din The Newsroom 3 diskete i ptt 5500 din. Besplatan katalog Siniša Žunić 9 proleterska 29 74000 Doboj tel 074/23--154

Prodajem za C64 interfejs za običan kasetofon (6500 din) Simons Basic (1500 din) Viza Vrite (1500 din) Easy Skript (1500 din)) Turbo programi 250 2002 2 Tape Fast, Abc Tel 011/162-132

Audio video komplet 2 muzički i porno programi Music Shop Sexkeid Sex Games II Partu Dirle i još 20 najnovijih uz iznenađenja sa kasetom i poštarinom 2100 din. Miodrag Vujković Bul revolucije 111 Beograd 011/403-852

Prodajem računar Commodore 64 sa diskom VC-1541 džojstik Qvickshot II i diskete (količine ograničene). Uz računar dobijete i literaturu. Tel 022/222-256 Aleksandar

Prodajem malo upotrebljavan Komodor +4 sa kasetofonom i 150 igara. Jordanović Predrag Sinđelićeva P+13/53 37000 Kruševac, tel 037/29-056

ELITE CRACKING CORPORATION
— Imamo sva najnovije programe
— 1 program = 50 din. (bez izuzetka)
— Komplet 45 prg + kaseta = 2500 din
— Izvanredna mogućnost pretplate
— Besplatan katalog i popust
— Tel 097/24-873 ili 097/22-540
Požarevačke 38 97000 Bitola ili 12 kladenci 11 97000 Bitola

Commodore 16 116 +4 — prodajem 15 novih igara za 1200 din. Na naručenih 45 igara dobijaš 15 na poklon. Branka Zulin Magistrala 105 58312 Podstrana. tel 058/651-453

Prodajem CPC-464 sa zelenim monitorom 500 igara i uslužnih programa džojstikom kvikšot 2 i mnoštvo literature na srpskohrvatskom. Matić Vladan I L Ribara 11319 Milošavac

C64. Dandy Camelot Warriors Skaterock, Tomahavk, Flash Gordon Tarzan Maradona, 1943 Feud Odyssey ... 20 igara kaseta i ptt = 1400 din. Rabić Boris Pere Dokića 2-C 71000 Sarajevo 071/647-730

C 64 Kripton soft vam nudi najnovije i starije hitove zagarantovanog kvaliteta. Cene programa od 50—70 din. Katalog besplatan. Milan Radojičić Dragana Jevtića 10 32300 G Milanovac, tel 032/711-971 (Mikan) ili 032/710-328 (Aco)

C64 najnoviji i najveći hitovi iz drugih oglasa po nižim cenama. Porudžbine snimam i u vašem prisustvu. Katalog besplatan. Pozovite na tel 011/122-731 Ivan Marović III bulevar 120, ulaz B 11000 Beograd

CBM-CLUB Osijek vam nudi Uslužni programi jezici i igre za C64/128. Moguća i razmena programa. Adresa CBM CLUB OSIJEK (Halas Alan) Vijenac Vi Suk-a 93/I tel 054/44-226 ili Dlavaški Zoran, Dunavska 27 54000 Osijek

Samousluga kod Pećol Birajte sami svoje programe pouzdajte se u svoj ukus Crnjanski Petar 4 juli 14/12 23000 Zrenjanin

Commodore 20 16 +4 64 128 programa ima 4500! Za VC-20 katalog je besplaten kao i za C-16 a ima novih paketa. Za C-64 mesečno proširujem ponudu za 300 din u markicama šaljem spisak. Đerman Šandor Rade Končara 23, 23000 Zrenjanin

DISKRETNI uslužni programi Wordstar 64 (kompletan sa uputstvom) Certificate Maker (Newsroom II) Oljgecad (sa uputstvom) 3D-design Print Master Ne morate imati IBM da bi koriste upotrebljavali računari Tražite ilustrovani katalog za opisima! Zoran Milosavljević, Splitska 4/4 34000 Kragujevac

Amstrad

AMSTRADIII BACK TO REALITY STAR FIREBIRD STRICH LUP HATERLAND RUNSTONE CLASSIC INVADERS ROBIN OF SHERWOOD TOP SONG Besplatan katalog tražite na 011/536-647 Borković Branko Partizanska 64 11090 Beograd

MAC SOFTWARE CPC 464
Najjeftiniji kompleti, 15 programa 1200 din Pojedinačno 130 din! Ovog meseca Komplet 21 Flat+ 2 Sigma 7 Uridium, Future Knight Arkanoid Heartland BMX Simulator Mario Krnjajić N Dragosavljević 82 55400 Nova Gradiška, 055/85-386 Katalog je besplatan

AMSTRADOVCI, veliko ljetno sniženje cijena kompletima programa u BROTHERS SOFT-u. Nemojte propustiti priliku koja vam se pruža Sve informacije u besplatnom ilustrovanom katalogu. Kolćžo Dževad i Nedžad Olnerska 28 71000 — SARAJEVO tel (071) 646-398

Amstradovci! SunSet-soft je za vas spremio najnovije hitove po najnižim cenema Feet (nogomet) Silent Service (simulacija podmornice) Heartland (Sorcery IV) Helichopper (legenda) Trailblazer (komentar nepotreban), BMX Simulator (vožnja biciklom) Acrojet (akrobatski mlaznjak). Ovo su za nas stare igre ali zato u besplatnom ilustrovanom kolor katalogu možete proveriti koje hitove imamo samo mi! Stojanović Dražen Vojvode Putnika 18-B 71000 Sarajevo 071/613-346

AMSTRAD CPC 464 6128 664
Nudimo vam veliki broj programa u kompletima. Na jednoj kaseti se nalazi od 14 do 24 programa a košta svega 2000 din+kaseta (1000)+ptt (500) Izabirati možete između 19 kaseta+zax+uslužni programi Naručivati možete i na disku na koji stane 6 programa po ceni od 7000 dinara za za komplet ili 500 dinara za pojedinačne programe Sve to možete naručiti isključivo na donjoj adresi FUTURESOFT POLJANSKI NASIP 30 61000 LJUBLJANA ili na telefon. 061-311-831

Amstradovci! Odaberite najbolje — DE LUXE SOFTWARE Imamo apsolutno sve nove programe (Mexico 86, Vera Cruz.) Sastavljate komplete pretplata, Poklovi, specijalne ponude hitna isporuka programi i u disk verziji Besplatan katalog Zoran Babović Rudarsko naselje 26 14233 Veliki Crljeni

AMSTRAD 6128, 664 464 (+ DISC)
— I DALJE NAJBOLJI IZBOR IGARA I USLUŽNIH PROGRAMA U JUGOSLAVIJI
— ISPORUKA ISKLJUČIVO NA DISKETAMA
— MINIMALNE CENE
— POPUSTI ZA STALNE KUPCE
— NARUČITE BESPLATAN KATALOG
Požurite! Preko 19 MB programa čeka na vas!
Nikolić Dragan, 14233 Vel Crljeni Kolonija 46

Amstradovci! Direktno iz Engleske smo nabavili Advanced Art Studio kao i sve najnovije igre (Night Lader Expres Rider) Na kaseti III ,3 disketi po izuzetno niskim cenama. Danilovski Denis Juri Gagarin 39A, Skopje 091/222-329 ili Acev Ivan Petar Pop Arsov 6 91000 Skopje, 091/234-128

Soft Box amstradovci svi koji ste zainteresovani za najnovije igre ne propustite priliku da za minimalne cene nabavite najnovije hitove Proverite naručite Isenović Dino Ace Navažilova bb 76250 Gradačac tel 076/87-028

Amstradovci Najnovije igre — Komplet 14 (Contanimation Yie-ar 2 Cobra Frost byte Xarg Komplet 15 (Star Glider Infrodroid Vixtora Jail Break Kralton) Posedujem ostalih 13 kompleta. Komplet sa kasetom i ptt troškovi 3000 din pojedinačno Plaćanje pouzećem Cvetković Ivan 16000 Leskovac A Dunjskog 17 tel 016/43-710

AMSOFT YU

AMSOFT YU CP/M Software predstavlja najnovije CP/M programe micro COBOL XLISP FORTH-83 Small-C (floating point) NEWCPM 83 k TURBO PASCAL ROS 3.3 FX-Character Generator CP/M Machine Code Input-Output Help WRITE HANG MAN CBASIC 80, EXBASIC OR DREW DR GRAPH CP/M Igre MEGAN3 AUSOPOLY BACARRAT PACMAN REVERSI 3D CLOCK CHESS Amsoft YU Spinčićeva 5 41000 Zagreb tel 041/315-476

Amstradovci! Venera Soft vam nudi jeftine programe (50—70) dinara u kompletima i pojedinačno Besplatan katalog. Stančić Dragan 4 jul 18/2, 23000 Zrenjanin tel 023/45-037

Prodajem Amstrad CPC464 sa zelenim monitorom + 36 kaseta sa softverom + literatura. Goran Strelić Strahinjića bana 2/7 18000 Niš 018/43-945

Danger soft vam predstavlja megahiteve iz Londona. Five s side Soccer — (Mach Dag 2). Thompson, Fu Kung in Las Vegas — (Karate) Silent Service — (igra za 1000) Trap Door — (pitajte comodorovce) Feud — (prvi na top listi) String Man2 (snagatori) Uskaroll! Two on Two (košarka) V O D (Vijetnam), Theatre Europe Southern Belle 5Th axe Koser Igor A. Cesarca 4 54400 Đakovo tel 054/841-812

Triton Soft je i ovog meseca izašao sa vrhunskim megahitovima. Sigma 7, Agent X, BMX simulator Iznenađenja! Nagrade! Besplatan katalog!!! Triton Soft Živka Žižića 6 84000 Bijelo Polje tel 084/24-803 ili 22-427

Hitno prodajem nov ocarinjen Amstrad CPC 464 sa zelenim monitorom Teofilski Jovan. JNA 222 Ruma 022/424-428 i 411-611

Amstrad superhitovi 16 programa 2800 dinara. Knightrider Marbles Cobra Kanamis Oule Great Escape Inforroid Miami Vice Tasnetos Star Glider Tarzan Feud Spacehakrler Elibon Pyjamarama 3 Zorad Flyepy Čobanov Branislav P Drapšina 53/1 21480 Srbobran tel 021/730-364 od 8—12 i 18—24 h

Kompleti CP/M i utility programa Komplet LANGUAGES FORTRAN PISTOL, JRT PASCAL, micro PROLOG Komplet TEXT WORDSTAR MAILMERGE PROSPELL ROTATE Komplet STAT AMSTAT 1—3 (statistički paket) Komplet Plus dBASE II SUPERCALC 2 WORDSTAR 3 34 ZIP SCK Komplet 2 2 MICROSCRIPT MICROPEN MICROSPREAD CP/M Utilities dbase II Utilities Architecture Utilities C-Archive, SPOOL XDRIVE 10 (Ram-disc) TURBO PASCAL Graphic TOOLBOS Poklon. CANBASE DATABASE Novi AMSDOS programi MASTERFILE III 6128 TASWORD 6128 YU TASSPELL MINI OFFICE 2 PROFI PAINTER Hardware proširenje 464 na 6128 (CP/M 3 0) RAMDISC 256 k EPROMI sa YU slovima za DMP-2000 i NLQ 401
Amsoft YU Spinčićeva 5, 41000 Zagreb tel 041/315476

IBM

Vlasnici IBMPCX/AT računara! Uputstva (latinica hrvatsko-srpski jezik) s primerima programa za red 1 CLIPPER kompajlerom za dBASE III — 10 hiljada dinara. Šaljem odmah po prijemu poštanske uplatnice. Dipl ecc Zlatko Gerenčir Matije Gupca 56 54551 Belišće. 054/81-679 (po podne)

IBM/PC programi Originalne i prevedene uputstva. Povoljno za Radne organizacije CHARLIE SOFT Borska 35, Lameia A 5/7 71210 Ilidža

NAJJEFTINIJE NAJEKSPEDITIVNIJE Za IBM PC/XT I KOMPATIBILNE RAČUNARE
Prodajem i menjam najbolji izbor programa dBASE III+ AUTO-CAD WORDSTAR LOTUS 123 OEM MULTIPLAN SYMPHONY Razni compajleri Utility progr i još 90-tak programa Sakes Antun 41040 Zagreb I Mišutinovića 34, tel 041/254-561

ATARI

PREVODI
Srpskohrvatski latinice

1 PROGRAMSKI	JEZIK
C	5800 din.
2 Atari St	
— Priručnik	3500 din.
— BASIC ST	3800 din
— LOGO ST	2700 din.

plus 700 din. za ptt. isporuka pouzećem
M Karabašević, Post restant 19210 Bor

Power without the price — programi za Atari 800XL/130XE (samo diskete) Preko 800 naslova. Ubrzano učitavanje Umerene cene Pouzdane usluge. Katalog 200 din Zvonko Atlija Zagrebačka 21 51000 Rijeka. tel 051/37-723

Ascii Soft. Imamo nove programe Shimera Rad Max, Scooter Achilles Ninja Crystalraider Povoljne cene Katalog besplatan. Pandurov Zoran, Đurđevska 33 23000 Zrenjanin tel 023/63-521

Minja Software Prodajem i menjam programe za Atari 800 XL Naručite besplatan katalog. Dimitrijević Predrag Sv Markovića 41/4, 19350 Knjaževac. 019/42-586

Atari XL/XE interface za običan kazetofon šeme 1000 kit 7000 a gotov i ispitan uređaj 9000 din Pouzdan u radu Smetiško Andrija Gosposvetska 67 62000 Maribor

ATARI ST — SOFTWARE
Najpovoljniji uslovi duplo jeftiniji programi cena od 300 do 1000 din Vip Platine Desktop Publish Text procesori baze podataka igre Katalog besplatan tel 021/587-025

BBC/ELECTRON

Solidisk 64 Kb Ram proširenje za BBC kompjuter prodajem Služi kao Rom table printer bafer i Ram disk. Dejan Čerović Kireviјeva 3/4 11030 Beograd tel 011/554-433

HARDWER

Prodajem nov printer Seikosha SP 1000CPC sa kablom za Amstrad CPC 2 traka i traktorom za 5om Tot Čaba. 24300 Bačka Topola Nad Jožefa 6 tel 024/713-979

Prodajem Z80CPU (2 200) D8039HLC (2 400) 80C51-64 (3 600) 8255-AC5 (2.400) 6802P (3 000) 680 SP (4 800) 6850P (2.200) 2716 (2 500) 2732 (2 600) 2764 (2 800) 27128 (3.000) 6264LP 15 (3.000) Kristali 4.000 6000 6 144 12 000 20.000 25 000 (800) Stabilizatori 78XX 79XX (400) Minimalna narudžba 5 000 din Za detaljan spisak pošalji adresiranu kovertu i 50 din Petar Bošnjaković Istarska 11 54400 Đakovo

Spektrumovci iskoristite sledeće usluge Servis Spectruma programiranje EPROM-a elektroničko hladilo za Spectrum Telefonirati oko 19h. Igor Bebić Vijenac M Pijade 35 54000 Osijek, tel 054/23-305

„Div" service Kvalitetno popravljam prodajem Sinclair Računare i Galaksiju. Ugrađujem proširenje memorije 16K na 48K reset tonski izlaz. Dipl ing Zoran Dimitrijević, 37000 Kruševac, 7 jula 7A tel 037/22-434

KOMPJUTER SERVIS
Spektrum komodor periferije — servisiram u vašem prisustvu. Telefon za dogovor 011/332-275 Ćosić Nenad Mišarska 11 Beograd

Hardware Servisiram kućne računare Prodajem proširenje 16K za ZX81 Eprom-e Joystick interface za 1 i 2 Palice I/O Port ZX Uprint III interface za Spectrum (RS 232 i Centronics sopstveni Eprom Štampanje teksta i grafike za Epson Epson kompatibilne Seikosha, Speedy i druge štampače) Dipl ing Branislav Karadžić 18000 Niš Branka Miljkovića 58/2 tel 018/328-488 od 17—20 h

Prodajem ZX Spectrum 128+' ocarinjen 26. 2 1987 sa uputstvom Trajković Slobodan, Lole Ribara 5/3 17500 Vranje tel 017/21-892

Literatura

IBM PC I KOMPATIBILNI RAČUNARI
PREVEDENA UPUTSTVA ZA PRG-
— dBASE III
— DBASE III
— FRAMEWORK
— LOTUS 1-2-3
— MS DOS 3 2
— WORDSTAR 3.24
— FINAL WORD 1 1
— TURBO PASCAL 3 0
— GW BASIC
SKRAĆENI PREVODI UPUTSTAVA
— AUTOCAD 60 str
— dBASE II 60 str
— dBASE III 55 str
Posjedujemo veći broj neprevedenih uputstava za upotrebne programe Mogućnost narudžbi za radne organizacije
Informacije na telefon
071/621-025 ili 071/455-562

COMMODORE – 64 PROFESIONALNI PREVODI PRIRUČNIK (1 700) PROGRAMMERS REFERENCE GUIDE (2 000) MAŠINSKO PROGRAMIRANJE (1 500). GRAFIKA I ZVUK (1 000) MATEMATIKA (1 200). DISK 1541 (1 000). Uputstva za usluzne programe SIMONS BASIC (700) PRAKTIKALK (800) EASY SCRIPT PASCAL MAE HELP-64+ VIZAWRITE STAT GRAF SUPER-GRAFIK po (600) MULTIPLAN (1 000) U kompletu 12 000.

SPECTRUM LITERATURA ZA RAD U MAŠINSKOM KODU MAŠINAC ZA POČETNIKE (1 600) NAPREDNI MAŠINAC (1 500) DISASEMBLIRANI ROM (2 000) DEVPAC 3 (600). U kompletu 4.500.

AMSTRAD-464 PROFESIONALNI PREVODI PRIRUČNIK — ukoričeno izdanje (2 500) LOCOMOTIV BASIC (1 400) MAŠINSKO PROGRAMIRANJE (1 400) Uputstva za usluzne programe DEVPAC, PASCAL, MASTERFILE TASWARD po (700). MULTIPLAN (1 000) U kompletu 7 500
„KOMPJUTER BIBLIOTEKA" Bate Jankovića 79 32000 Čačak, tel 032/30-34

Profesionalno urađen prevod knjige „Practical Programs for the Amstrad CPC464 Cena 3000 din. Živa Žebeljan Đura Đakovića 61, Pančevo 013/47-580

LITERATURA
Za sve koji se ozbiljno bave računarima neophodna je i dobra knjiga. COMET software Vam nudi originalnu LITERATURU na ENGLESKOM jeziku
— knjige za vaš računar
ZX SPECTRUM QL COMMODORE BBC, APPLE
— Mikroprocesori
— Programski jezici
— Operativni sistemi
— Obrada teksta
— Servisni priručnici
GARANCIJA ZA SVE VRSTE USLUGA
MILOVANOVIĆ LJUBIŠA, Petra Lekovića 57 11080 BEOGRAD tel 011/668007 posle 17 h 087008

Razno

Servisiram kvalitetno i jeftino računare Spektrum Kupujem ispravan Spektrum (do 70 000 din) ili neispravan (do 40 000) ovisno o stanju i kompletnosti Petar Bošnjaković Istarska 11 54400 Đakovo

Novo! Regenerišem ribone svih tipova matričnih printera originalnih boja Kvalitetan otisak. Šikić Željko Mirka Jovanovića 8 71000 Sarajevo 071/539-600 od 16 sati

PRODAJEM RAČUNAR PHILIPS MSX-VG 8020. tel 021/333-540 zvati posle 15 časova

Građevinari nudimo vam vrhunske programe za IBM PC Difuzija vodene pare (JUS U J5 600) i izračun kanala sa provetravanja Tražite katalog još danas! AP Software V Murglah 70 61000 Ljubljana

MBM
SERVIS PERSONALNIH RAČUNARA I PRATEĆE OPREME SERVISIRA
— SPEKTRUM
— QL
— KOMODOR
— AMSTRAD
— IBM PC KOMPATIBILCI
— ELEKTRONSKE PISAĆE MAŠINE
— STONE KALKULATORE
Milan Beronja Balzakova 8 Novi Sad tel 021/369-463
od 9—15h

Vic 20 Vic 20! Programi Garancija 3 godine! Snimci iz memorije Kompleti pojedinačno Kvalitet! Ima programa i za C64 Pojedinačno samo 50 din Uverite se! Još malo pa će biti programa za proširenja za VIC 20 Pukovi besplatni Samo preko telefona (071/212-366) Javite se! Za učlanjivanja potrebno je poslati svoje lične podatke isključivo u pismu Tada na svaki program dobijate 5 besplatnih (po želji) Pozdrav Isaković Denis Muhameda Džudže 33/2 stan 11 71000 Sarajevo

SPEKTRUMOVCI!!! PAŽNJA!!!
Najnoviji programi!!! Komplet 1100 dinara pojedinačno 200 dinara program Komplet 55 USKORO!!!
Komplet 54 INDOOR SPORTS (AIR HOCKEY, DARTS PING PONG BOWLING) SENTINEL (87% CRASH) RAMON RODRIGES SWORDS OF BANE TRAP ROAD RACER
Komplet 53 SABOTEUR 2 HEAD OVER HEELS ACEDEMY (TAU CETI 2), WHITE HEAT SILLYCON WAR STAR RUNNER NUCLEAR COUNT DOWN TERROR OF DEEP
Komplet 52 ENDURO RACER NEMESIS ARMY MOVES A W MONTY BIG TROUBLE LEGIONS OF DEATH CHORT CIRCUIT 1 2 STAR RIDERS 2 KRAKOUT
PREDRAG DENADIĆ O Kazaklajića 33 14220 LAZAREVAC tel 011/811-208

Prodajem šeme za satelitsku antenu TOKI-voki dometa 15—20 km Razdelnik i Bubu (minijaturni radio-predajnik veličine kutije šibice) Kantoci Ivan Nevice Čerovića 15 78000 Banjaluka 078/30-940

Prodajem Sinclair QL + Monitor (mono žuti) + Literatura + programi (Lattice c compiler Macro Assebler) tel 069/78-406 (20—22 h)

Prodajem neupotrebljavani Spectrumov Light-Pen + original kasetni program + uputstva Prodajem i nov neupotrebljavan japanski stereo Walkman Unicef SZ 10 Šaljite ponude Pejčinovski Zoran Beogradska 47a 97000 Bitola tel 097/42-241

Prodajem Printer 1526 i Disk 1541 za Komodor 64 Mitrević Slavko Svetozara Markovića 76 34000 Kragujevac tel 034/46-452

ELECTRONIC — SERVIS ZA POPRAVKU KUĆNIH RAČUNARA
Vršim kvalitetne popravke i prepravke računara Sinclair ZX, Commodore svih tipova, Galaksija i Amstrad sa garancijom na izvršene radove. Meletin Radovan 23330 Novi Kneževac, Trg Maršala Tita br 10. tel 023/541-002 od 15 do 18 h.

Kupujem IC 8255 PIO CD 4066 CD 4040 (2 kom) SN 74LS25 (2 kom) Majdandžić Suzan Ul Mala 129, 55400 Nova Gradiška tel 065/65-512

Prodajem Eprom 27128-A = 7 800 - Dram 4164/4116 = 4 000 Z80CPU = = 3 500 Z80A CPU = 4 500 SN74 LS00/32/157 = 750 Računar Laser 2101BKB = 65 000 Tel 022/74-244 posle 15 sah

Prodajem diskete 5 25 DSDD
Tel 041/428-574 ili 041/413 731

P.N P. electronic

Jeretova 12 58000 Split, tel (058) 589-867
radnim danom od 9—14 i 17—19 subotom 9—13
popravci izrada uređaja rezervni dijelovi potrošni materijal diskete literatura usluge savjeti besplatni katalozi za.

SPECTRUM	COMMODORE
PALICE (JOYSTICI)	PALICE (JOYSTICI)
KEMPSTON JOYSTIC INTERFACE	EPROM MODULI DO 0.5 MB (64 Kb)
DVOSTRUKI JOYSTIC INTERFACE	EPROM PROGRAMATOR
SVJETLOSNA OLOVKA	BRISAČ EPROMA
EPROM PROGRAMATOR	SVJETLOSNA OLOVKA
CENTRONICS PRINTER INTERFACE	CENTRONICS PRINTER INTERFACE
MEGAROM (EPROM MODUL)	MODEM ZA JUMBO
P.N.P. ROM (PREPRAVLJENI ROM)	RESET TIPKA
PROŠIRENJE MEMORIJE 16—48K (80)	VIDEO/AUDIO KABL ZA MONITOR

ATARI ST 260/520/1040

PROŠIRENJE MEMORIJE 1-2-4 M NA KARTICI BEZ LEMLJENJA JEDNOSTAVNO TOS U EPROMIMA — ENGLESKI NJEMAČKI ENGLESKO-NJEMAČKI I YUGO TV MODULATOR EPROM PROGRAMATOR CENTRONICS KABL ZA ŠTAMPAČ FAST BASIC KARTRIDŽ SA KOMPAJLEROM YU EPROMI ZA ŠTAMPAČE SAT MINIJATURNI DVOSTRANI DISK SA UGRAĐENIM ISPRAVLJAČEM U KUTIJU VELIKI IZBOR KVALITETNE LITERATURE I PROGRAMA POPRAVCI I SERVIS

Sa vama i u avgustu

„Računari" ne preskaču
nezgodne letnja mesece
Sledeći, avgustovski broj
izlazi iz štampe krajem jula
A u njemu,
kao što i priliči najtoplijem mesecu u godini,

KATALOG
PC SOFTVERA

sa jezgrovitim opisom najkapitalnijih programa
za IBM PC XT/AT i kompatibilnu bratiju
iz pera Arifa Agovića
i

RAZBARUŠENI
SPRAJTOVI

na povećanom broju strana.

Sam svoj loto programer

Nevolje sa susedima

Ako igrate loto, verovatno ste se još od „Računara" broj 25 uključili u razvijanje programa LOTO UNV (Loto univerzal), pa znate kakva vam se prilika pruža da napišete svoj program za izradu loto-sistema. Sve što treba da uradite to je da prepišete standardni deo programa (iz broja 25 ili 27, svejedno) i da mu dodate rutine koje sami odaberete iz prethodnih i nekoliko sledećih brojeva

Jedino ograničenje u komponovanju programa od modula koje objavljujemo jeste da prva odabrana rutina počne od adrese TEST Sve ostale možete jednostavno nanizati iza nje Raspored rutina nije bitan za korektan rad programa možete ih složiti kako kog hoćete Jedino treba da imate na umu da je zbog brzine uputno da najpre unesete rutine koje se izvršavaju za kraće vreme — time se sporijim rutinama oduzima šansa da ispituju znatan broj kombinacija

Nezgodni susedi

Znači rutine koje odaberete treba da smestite između standardnih rutina GENER i MEM I to je sve

Ukoliko ste propustili priliku da se ranije pozabavite ovom rubrikom „Računara", videćete da još nije kasno da se uključite Prvi nastavak je izašao u broju 25 a vi komplet „Računara" verovatno čuvate ili ga možete nabaviti

Jeste li već imali nevolje sa susednim brojevima? Odlučite se da ih isključite iz sistema — a oni se pojave u dobitnoj kombinaciji i obratno Na kraju pomislite da se ništa ne može predvideti s bilo kakvom sigurnošću

Da li je zaista tako? Pogledajte rezultate koje vam pruža LOTO GRAF Šta se dogodilo u 50 uzastopno odigranih kola (prvih 20 iz ove i poslednjih 30 iz prošle godine)? Možete li izvući zaključak šta će se verovatno desiti u sledećih deset ili dvadeset kola? Naravno da može.

Ostalo je znači da od ponuđene četiri mašinske rutine odaberete onu kojom ćete postići najveći stepen verovatnoće za dobitak

Loto i statistika

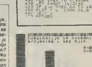
Mladi matematičar ima puno znanja, ali malo životnog iskustva Posmatra ulicu kroz prozor svoje sobe i kladi se sa prijateljem da će među prvih sto prolaznika biti bar jedna žena Verovatnoća je na njegovoj strani, zar u gradu nije skoro podjednak broj muškaraca i žena?

Ali na lazi vražel U tom trenutku začuje se bleh-muzika Mladić i njegov prijatelj ugledaju kroz prozor bataljon vojnika u maršu, predvođen bučnim orkestrom!

Ova anegdota iz knjige ruskog matematičara J I Perelimana „Zanimljiva matematika" upozorava na činjenicu koji svaki igrač lota mora imati na umu statistika se može uspešno primaniti prilikom pravljenja sistema ali ne u igri za samo jedno kolo Da se mladi matematičar više puta uzastopno kladio ukupan rezultat bi svakako bio povoljan za njega Tako je i u lotu Nema smisla baviti se statistikom ukoliko nije u pitanju dugoročna i sistematska igra

```
 10 REM -----------------------
 20 REM    L O T O   G R A F
 30 REM -----------------------
 40 DIM b(7): DIM c(7)
 50 PRINT PAPER 6;AT 0,4;"L O T O  G R A F  za 50 kola"
 60 PRINT AT 1,4;"Kombinacije sa susednim";AT 2,4;"brojevima i bez njih"
 70 REM -----------------------
 80 REM  nalazenje sus. brojeva
 90 REM -----------------------
100 FOR i=1 TO 50
110 FOR j=1 TO 7: READ b(j): NEXT j
120 LET s=0
130 FOR j=1 TO 6
140 IF b(j)=b(j+1)-1 THEN LET s=s+1
150 NEXT j
160 LET s=s+3
170 LET c(s)=c(s)+1
180 NEXT i
190 REM -----------------------
200 REM          obrada podataka
210 REM   pozivanje potprograma
220 REM -----------------------
230 PRINT AT 4,23;"A=B+C";AT 5,23;"C=D+E+F+G"
240 LET c(1)=c(2)+c(3)+c(4)+c(5)+c(6)
250 FOR n=7 TO 4 STEP -1
260 LET c(n)=c(n-1)
270 NEXT n
280 LET c(3)=c(4)+c(5)+c(6)+c(7)
290 LET a=65: LET t=0: LET x=0
300 LET s=0
310 FOR n=1 TO 7
320 LET kr=c(n): GO SUB 510
330 PRINT AT 19,t;CHR$ a;AT 20,t;kr;AT 21,t;p$
340 GO SUB 420
350 LET t=t+4: LET a=a+1: LET x=x+32
360 IF n=1 OR n=3 THEN LET t=t+2: LET x=x+16
370 NEXT n
380 PAUSE 0: COPY : STOP
390 REM -----------------------
400 REM                 crtanje
410 REM -----------------------
420 FOR y=24 TO kr*3+24
430 LET z=INT (y/3)*3
440 IF z=y THEN GO TO 460
450 PLOT x,y: DRAW 24,0
460 NEXT y
470 RETURN
```

```
 480 REM -----------------------
 490 REM  izracunavanje procenta
 500 REM -----------------------
 510 LET p=kr*50+100
 520 LET p$=STR$ p+"%"
 530 RETURN
 540 REM -----------------------
 550 REM     dobitne kombinacije
 560 REM -----------------------
 570 DATA 1,2,10,13,22,29,39
 580 DATA 6,7,17,18,24,30,35
 590 DATA 1,2,7,18,21,25,27
 600 DATA 1,11,17,22,26,30,31
 610 DATA 16,18,21,27,28,35,39
 620 DATA 3,14,23,27,32,35,36
 630 DATA 2,9,15,23,31,38,39
 640 DATA 1,9,11,17,29,32,38
 650 DATA 1,5,10,22,25,26,28
 660 DATA 6,9,17,22,30,37,38
 670 DATA 1,6,8,14,30,35,38
 680 DATA 1,3,14,20,23,28,34
 690 DATA 1,2,6,10,12,14,28
 700 DATA 12,14,15,18,20,27,37
 710 DATA 1,4,9,13,19,35,37
 720 DATA 3,8,16,18,19,30,31
 730 DATA 3,12,17,24,32,35,39
 740 DATA 1,3,15,22,23,33,39
 750 DATA 4,12,14,16,26,30,33
 760 DATA 6,11,19,21,22,29,30
 770 DATA 13,16,22,27,34,36,39
 780 DATA 4,8,16,24,36,37,39
 790 DATA 1,8,12,14,21,24,27
 800 DATA 3,7,15,21,23,26,31
 810 DATA 4,11,19,22,26,29,34
 820 DATA 2,11,15,18,28,32,33
 830 DATA 1,9,15,27,31,37,38
 840 DATA 6,9,10,15,24,27,39
 850 DATA 4,5,11,29,31,38,39
 860 DATA 2,4,23,28,30,31,36
 870 DATA 1,12,14,17,23,28,31
 880 DATA 10,12,13,15,18,31,37
 890 DATA 2,7,14,15,30,34,37
 900 DATA 6,23,31,32,33,35,38
 910 DATA 2,4,11,17,25,32,33
 920 DATA 5,9,11,26,29,36,38
 930 DATA 1,5,25,26,32,35,36
 940 DATA 4,8,15,16,24,25,35
 950 DATA 10,14,18,27,33,34,37
 960 DATA 14,18,17,20,21,31,34
 970 DATA 9,25,28,30,32,35,38
 980 DATA 1,4,21,23,24,29,39
 990 DATA 10,11,16,17,30,34,31
1000 DATA 4,9,10,11,19,29,31
1010 DATA 1,5,8,23,30,33,36
1020 DATA 5,7,9,34,35,38,39
1030 DATA 2,4,11,25,26,30,37
1040 DATA 5,8,23,30,32,33,39
1050 DATA 6,9,15,23,31,32,33
1060 DATA 8,9,12,16,23,28,31
```



Stub A prikazuje ukupan broj kombinacija B su kombinacije bez susednih brojeva, a C kombinacije s njima. D su kombinacije s jednim parom susednih brojeva, E — sa dva para, F — sa tri para i G — sa četiri para

Blizanci i oko njih

Rutina „Maks 3 uzastopna" eliminiše iz sistema sve kombinacije koje imaju više od tri uzastopna broja Neće dozvoliti na primer, da se pojavi kombinacija 1 3, 4, 5, 6, 10, 12

Jednostavnom redukcijom nekih naredbi koju možete lako da uočite dobićete rutinu koja dozvoljava najviše dva uzastopna broja (jednom ili više puta u kombinaciji), a zatim i rutinu koja ih potpuno isključuje

Poslednja rutina „Blizanci" (tako igrači lota nazivaju par susednih brojeva) eliminiše

```
1510 ;------------------------------
1520 ;    MAKS. 3 UZASTOPNA
1530 ;------------------------------
1540 TEST   LD   HL,BKO
1550        LD   B,DKO-3
1560 pb1    LD   A,(HL)
1570        INC  HL
1580        INC  HL
1590        INC  HL
1600        INC  A
1610        INC  A
1620        INC  A
1630        CP   (HL)
1640        JP   Z,GENER
1650        DEC  HL
1660        DEC  HL
1670        DJNZ pb1
```

```
1510 ;------------------------------
1520 ;    MAKS. 2 UZASTOPNA
1530 ;------------------------------
1540 TEST   LD   HL,BKO
1550        LD   B,DKO-2
1560 pb1    LD   A,(HL)
1570        INC  HL
1580        INC  HL
1590        INC  A
1600        INC  A
1610        CP   (HL)
1620        JP   Z,GENER
1630        DEC  HL
1640        DJNZ pb1
```

```
1510 ;------------------------------
1520 ;    BEZ UZASTOPNIH
1530 ;------------------------------
1540 TEST   LD   HL,BKO
1550        LD   B,DKO-1
1560 pb1    LD   A,(HL)
1570        INC  HL
1580        INC  A
1590        CP   (HL)
1600        JP   Z,GENER
1610        DJNZ pb1
```

```
1510 ;------------------------------
1520 ;    BLIZANCI
1530 ;------------------------------
1540 TEST   LD   HL,BKO
1550        LD   C,0
1560        LD   B,DKO-1
1570 pb     LD   A,(HL)
1580        INC  HL
1590        INC  A
1600        CP   (HL)
1610        JR   NZ,1b
1620        INC  C
1630 1b     DJNZ pb
1640 ;
1650        LD   A,C
1660        CP   2
1670        JP   NC,GENER
```

sve kombinacije koje imaju dva ili više parova blizanaca. Ako iza naredbe CP u liniji 1660 ukucate broj 3, program neće propuštati kombinacije sa 3 i više parova blizanaca — i tako dalje.

Ako želite kombinacije isključivo sa dva para blizanaca, JP NC u liniji 1670 zamenite naredbom JP NZ. I u ovom slučaju promenom broja u liniji 1660 možete dobiti ono što želite.

Na kraju treba da raščistite još dve dileme.

Jesu li za vas tri broja zaredom (na primer 3, 4, 5) u isto vreme i dva para susednih brojeva (3—4 i 4—5) ili nisu? Igrači loto na ovo različito gledaju. Ako nisu a dozvoljavate dva ili više parova blizanaca, pored rutine „Blizanci" unesite u program i rutinu „Maks 2 uzastopna". Redosled i u ovom slučaju nije bitan. Ako dozvoljavate najviše jedan par blizanaca, to jest ako u liniji 1660 ostavite CP 2, onda, razumljivo, druga rutina nije potrebna.

Druga dilema se odnosi na shvatanje šta su u lotu susedni brojevi. Za neke igrače to su susedni brojevi u sistemu. Na primer, ako u sistemu imate brojeve 3 i 6, i nijedan broj između njih, onda su 3 i 6 susedni brojevi u sistemu. Drugi igrači pak smatraju susednim brojevima samo one koji se u prirodnom nizu nalaze jedan do drugoga. Ukoliko se opredelite za prvo shvatanje, rutina iz ovog broja će vam pomoći da dobijete tačno ono što želite. Ako zastupate suprotno gledište, biće vam potrebna još jedna rutina, za zamenu brojeva pre pravljenja sistema. Ali, o tome u nekom od narednih brojeva.

Žarko Vukosavljević

Prenos programa sa kasete na disk (2)

Iz više delova

U „Računarima 26" opisali smo tehnike prenošenja programa sa kasete na disk, ali one se mogu primeniti samo ako programi nisu duži od 42830 bajtova. Dugački programi se mogu presnimiti na dva načina: deljenjem na dva ili više delova, tako da se višak preko 42830 bajtova može smestiti u video memoriju i kasnije pomoću opcije LDIR vratiti na svoje mesto, i alociranjem zaglavlja.

Prvi korak je da na već opisani način (vidi „Računare 26") ukrademo iz Bloka 1 podatke o vrednostima A, HL i DE registara neposredno pre poziva sistemskog potprograma CAS-READ (CALL #BCA1). Naka sve to izgleda npr. ovako:

```
ld a, #16
ld hl, #1000
ld de, 45000
call #bca1
```

Kada bismo ovakav program pokušali da presnimimo na neki od načina koji su prikazani u prošlom nastavku, program ne bi radio. Zašto? Razloga ima više, a mi ćemo vam skrenuti pažnju na dva najvažnija.

Kada se program učitava sa diskete, zaglavlje sa svim potrebnim podacima smešta se počevši od adrese &A756, što je veoma nisko — naš program bi ga tokom učitavanja jednostavno „pojeo" svojom dužinom i program ne bi znao odakle da se startuje. Drugi razlog je da program koji je duži od 47 K „upliva" u JUMP BLOCK svojim zadnjim delom. To „uplivavanje" najčešće izaziva haos u samom računaru. Da bi se to izbeglo, program bez zaglavlja treba podeliti na dva dela na sledeći način:

```
ld a, #16
ld hl, 1000
ld de, 45000
call #bca1
ld c, 7
call #bcce
ld b, l
ld hl, qqqq, adresa naredbe
ld c, c
ld de, pppp, adresa bufera
od 2 K, pppp=31000
call #bc8c
ld hl, 1000
ld de, 30000
ld a, 2
call #bc98
call #bc8f
ld c, c
```

i pustiti da se „headerless" učita od početka do kraja i snimi njegov prvi deo na disketu pod imenom I.bin. Zatim treba premotati kasetu na početak i primeniti isti postupak ali ovoga puta uz pomoć sledećeg programa:

```
ld, a, #16
ld hl, 1000
ld de, 45000
call #bca1
ld c, 7
call #bcce
ld b, l
ld hl, qqqq, adresa naredbe
ld c,d
ld de, pppp, adresa bufera
od 2 ko pppp 28000
call #bc8c
ld hl, 31000
ld de 15000
ld a, 2
call #bc98
call #bc8f
ld c d; ASCII kod za „J"
```

Ovaj deo „headerless" treba snimiti pod imenom J.BIN.

Na ovaj način podelili smo program na dva dela koji sada više nisu bez zaglavlja. Ako želimo da podelimo program na više delova, postupak je identičan ovom, samo za tako nešto uglavnom nema potrebe. Da bismo učitali ova delova, potrebno je napraviti loader u bejziku ili mašincu na sledeći način:

```
10 mode 0
20 openout „d"
30 memory 999
40 load „I" 1000
50 load „J" 49152
60 for x= #bf00 to #bf00+11
70 read d: poke x, d: next
80 call # bf00
90 call ENTRY ADRESS
1000 data & 21, &00, &c0, &11
&18 &79, &&1, &98 &3a
&ed, &b0 &c9
```

Linije 60—90 mogu se zameniti ekvivalentnim mašinskim programom.

```
org #bfoo
ld hl #cooo
ld da 7918
ld bc 3e98
ldir
jp kkkk, kkkk je ENTRY ADRES
```

Ovim bismo završili priču za slučaj kada pribegavamo podeli „hedarlesa" Ovaj način trebalo bi zapamtiti, jer se i danas (mada retko) neki komercijalni softver zaštićuje samo osnovnom dužinom što se može jednostavno aliminisati

Relocirano zaglavlje

Postoji, medutim, način da se izbegne nedostatak ova tehnika kad program prelazi preko sopstvenog zaglavlja Trebe jednoatavno pravariti računar tako da se zaglavlje učita ne neku drugu adresu a ne na #A756 Ta adresa na trebe da bude fiksna, već može da se nađe bilo gda na slobodnom mestu u memoriji Mi vam preporučujemo adresa #AFOO i #BFOO jer zaglavlja tu obično nikoma ne smeta Nedostatak ovog metoda je da se program koji je snimljen na ovaj način ne može ni jednim COPY programom vratiti nazad na traku zbog toga što smo alocirali zaglavlja Ipak, za vlasnika diskete je ovaj način najprihvatljiviji jer se ovako štedi najmanje dva kilobajta na disketi a može se postići i veća ušteda sažimanjem programa Praktično to sve izglede ovako Treba premotati traku na početak programa bez zaglavlja i startovati sledeći program

```
10 ld a, 16
20 ld hl 1000, od adrese 1000
30 ld de 45000; u dužini 45000
40 call #bcal učitava program
bez zaglavlja
50 ld c 7
60 ld hl #bfoo+63
alocira zaglavlje na #bfoo i
izvršava I DISC
70 ld de, # bfoo
60 call #bcce
90 ld b, l
100 ld hl, xxxx, adresa za
naredbu ld c,c
110 ld de #cooo
120 call #bc8c
130 ld hl, 1000
140 ld de 45000
150 ld bc ENTRY ADRESS
160 ld a,2
170 call #bc98
160 call #bc8f
190 ld c,c, ASCII kod za ,I'
```

Ovde trebe posebno obratiti pažnju na linija 50—80 jer se u njima izvršava alokacija zaglavlja se #A756 (podrazumavene adresa) na BFOO, kao i na liniju 150, gde se U BC smešta ENTRY, ADRESS koja je u ovom slučaju obavezna

Ovim bi bila završena naša priča o prenosu programa bez zaglavlja sa kasete na disketu Za kopiranje ekstremno dugih programa trebe primeniti tzv MAXBYT operaciju Tada program počinjamo da učitavamo od #COOO; kada do[illegible] do #ffff pralazi na nulu i ida opet iz početka Medutim u ovom slučaju ne možemo komunicirati sa JUMP BLOCK-om (jer su nam pokrivene RST naredbe) već direktno sa romovima O toma kao i konverziji disk-disk u nakom sledećem nastavku

Žaljko Ciganović

Amstrad / crtanje kruga

Krug za pet stotinki

Danas programari izvlače naverovatno brzu grafiku iz šesnaestobitnih računara, ali velika brzina nije nedostupna ni na osmobitnim mikraćima. Amstradov procesor poseduja veoma moćne naredbe za rad sa grafikom: 16 bitna aritmetika i SET, RES i BIT skupovi naradbi. Njima se može direktno paliti bit u bajtu, tj. tačka u MODE 2. Zahvaljujući ovim pogodnostima, uz primanu dobrog algoritma na „amstradu" se može postići veoma brza grafika.

Rutina koju smo pripremili za ovaj broj crta 199 krugova (poluprečnika od 1 do 199) u petlji za oko 13,4 sekundi Merenja je izvršeno TEST rutinom iz bejzik primera, prilikom merenja omogućiti interapt) Vreme potrebno da se jedan krug nacrta je oko 0 06 sekundi (efektno 0 05 s, brže nego što ljudsko oko može da registruje), I pri toma se ništa ne gubi na kvalitetu kružnice! Rutine se izvodi različitom brzinom zavisno od radijusa kružnice Ova rutina je oko 1,8555 puta brža od nejbrže ekvivalentne rutine ranije objavljene u domaćim časopisima (za amstrad', skoro sve su objevljene u Računarima") Po brzini se opasno primakla i CIRCLE rutini GEM-a (testiran GEM PAINT ‚šnajdar PC1512" 8086 16 bita, 8 MHz, 64 KVIDEO mamorija) a po kvalitatu nacrtane kružnica rutina iz primera je pretekla GEM za dva reda veličine

Brzine rutine se može izmeriti bejzik programom iz primera (naravno, iz rutine izbeciti DI naredbu) Po utrošenom vremenu je, ipak bolje od sličnih rutina na osmobitnim računarima koje ne koriste tabela U

```
10 MODE 2
20 REM       Bejzik primar
30 REM        Test brzine
40 DI
50 a=TIME
60 CALL 21000
70 b=TIME-a
80 PRINT"Sekundi"b/300,"prosacno"b/300/199
90 EI
100 END
      10 MODE 2
      20 DE=320 HL=200.IX=100 IY=50
      30 '
      40 '          BASIC
      50 '      ELIPSE rutine
      60 '
      70 DI
      80 x=0:y=1
      80 a=0 999847695
      100 b=0 017452408
      110 d=1 99969539
      120 xx=b'yy=a
      130 MOVE DE,HL+IY
      140 FOR x%=0 TO 360
      150 DRAW DE+IX*x,HL+IY*y
      160 xy=d*xx-x'yx=d*yy-y
      170 x=xx y=yy xx=xy.yy=yx
      180 NEXT
      190 EI
      200 END
```

CIRCLE rutini su korišćene super brze PLOT i A-HL rutine koje su ranija objavljene u „Računarima' (‚Crtanja na mašincu"), i detaljno objašnjene Ulazni ragistri su standardni i određuju x, y koordinatu centra i radijus (poluprečnik) kruga

DE — x koordinata centra kruznice (0<=x<=639),

HL — y koordinata centra kruznice (0<=y<=398)

BC — r radijus kruznice (0<=r<=199)

Sama CIRCLE rutina počinje od 200 linija a potprogram od 10 do 160 linije predstavlja petlju koja demonstrira brzinu rutine Rutina je čista što znači da ne formira nikakve tabele i ne zauzima dodatnu memoriju.

Ova rutina je brža od ostalih zato što se jedno izračunavanje koordinata grafičke linije izvodi za dve tačke a u jednom prolazu se nacrta osam tačaka Izbečeni su realni brojevi — koriste se samo celi, manji od 65535 koji se mogu obrađivati procesorskom aritmetikom Na žalost brzina je uzela svoj danak pa rutina ima i nekih mana kružnica ne sme da izađe van ekrana (ali se to može lako prevazi-

```
Pass 1 errors: 04

                  10 ;d
21000             20        org  21000
21000             30        ent  $
21000 010000      40        ld   bc,0
21003 C5          50 ST:    push bc
21004 114001      60        ld   de,320
21007 21C800      70        ld   hl,200
21010 CD2152      80        call CIRCLE
21013 C1          90        pop  bc
21014 03         100        inc  bc
21015 21C700     110        ld   hl,199
21018 B7         120        or   a
21019 ED42       130        sbc  hl,bc
21021 C8         140        ret  z
21022 09         150        add  hl,bc
21023 18EA       160        jr   ST
                 170
                 180
                 190
21025 F3         200 CIRCLE: di
21026 ED537453   210        ld   (XX),de
21030 227653     220        ld   (YY),hl
21033 50         230        ld   d,b
21034 59         240        ld   e,c
21035 210000     250        ld   hl,0
21038 010000     260        ld   bc,0
21041 227053     270        ld   (HHL),hl
21044 EB         280        ex   de,hl
21045 226E53     290        ld   (EEE),hl
21048 C5         300 START: push bc
21049 2A7053     310        ld   hl,(HHL)
21052 ED5B7653   320        ld   de,(YY)
21056 19         330        add  hl,de
21057 CB3C       340        srl  h
21059 CB1D       350        rr   l
21061 3EC7       360        ld   a,199
21063 95         370        sub  l
21064 CD5553     380        call A_HL
21067 227253     390        ld   (HP),hl
21070 2A6E53     400 ST1:   ld   hl,(EEE)
21073 EB         410        ex   de,hl
21074 2A7453     420        ld   hl,(XX)
21077 19         430        add  hl,de
21078 EB         440        ex   de,hl
21079 CD3353     450        call PLOT
21082 F5         460        push af
21083 D5         470        push de
21084 2A6E53     480        ld   hl,(EEE)
21087 EB         490        ex   de,hl
21088 2A7453     500        ld   hl,(XX)
21091 B7         510        or   a
21092 ED52       520        sbc  hl,de
21094 EB         530        ex   de,hl
21095 CD3353     540        call PLOT
21098 D5         550        push de
21099 2A7053     560        ld   hl,(HHL)
21102 EB         570        ex   de,hl
21103 2A7653     580        ld   hl,(YY)
21106 A7         590        and  a
21107 ED52       600        sbc  hl,de
21109 CB3C       610        srl  h
21111 CB1D       620        rr   l
21113 3EC7       630        ld   a,199
21115 95         640        sub  l
21116 CD5553     650        call A_HL
21119 227253     660        ld   (HP),hl
21122 D1         670        pop  de
21123 CD5153     680        call TT
21126 D1         690        pop  de
21127 F1         700        pop  af
21128 CD4853     710        call TV
21131 2A6E53     720        ld   hl,(EEE)
21134 EB         730        ex   de,hl
21135 2A7053     740        ld   hl,(YY)
21138 19         750        add  hl,de
21139 CB3C       760        srl  h
21141 CB1D       770        rr   l
21143 3EC7       780        ld   a,199
21145 95         790        sub  l
21146 CD5553     800        call A_HL
21149 227253     810        ld   (HP),hl
21152 2A7053     820        ld   hl,(HHL)
21155 EB         830        ex   de,hl
21156 2A7453     840        ld   hl,(XX)
21159 19         850        add  hl,de
21160 EB         860        ex   de,hl
21161 CD3353     870        call PLOT
21164 F5         880        push af
21165 D5         890        push de
21166 2A7053     900        ld   hl,(HHL)
21169 EB         910        ex   de,hl
21170 2A7453     920        ld   hl,(XX)
21173 B7         930        or   a
21174 ED52       940        sbc  hl,de
21176 EB         950        ex   de,hl
21177 CD3353     960        call PLOT
21180 D5         970        push de
21181 2A6E53     980        ld   hl,(EEE)
21184 EB         990        ex   de,hl
21185 2A7653    1000        ld   hl,(YY)
21188 A7        1010        and  a
21189 ED52      1020        sbc  hl,de
21191 CB3C      1030        srl  h
21193 CB1D      1040        rr   l
21195 3EC7      1050        ld   a,199
21197 95        1060        sub  l
21198 CD5553    1070        call A_HL
21201 227253    1080        ld   (HP),hl
21204 D1        1090        pop  de
21205 CD5153    1100        call TT
21208 D1        1110        pop  de
21209 F1        1120        pop  af
21210 CD4853    1130        call TV
21213 C1        1140        pop  bc
21214 2A6E53    1150        ld   hl,(EEE)
21217 EB        1160        ex   de,hl
21218 2A7053    1170        ld   hl,(HHL)
21221 A7        1180        and  a
21222 ED52      1190        sbc  hl,de
21224 23        1200        inc  hl
21225 29        1210        add  hl,hl
21226 09        1220        add  hl,bc
21227 EB        1230        ex   de,hl
21228 2A7053    1240        ld   hl,(HHL)
21231 29        1250        add  hl,hl
21232 23        1260        inc  hl
21233 09        1270        add  hl,bc
21234 E5        1280        push hl
21235 D5        1290        push de
21236 CB7C      1300        bit  7,h
21238 CAFF52    1310        jr   z,DE
21241 7D        1320        ld   a,l
21242 2F        1330        cpl
21243 6F        1340        ld   l,a
21244 7C        1350        ld   a,h
21245 2F        1360        cpl
21246 67        1370        ld   h,a
21247 CB7A      1380 DD:    bit  7,d
21249 2808      1390        jr   z,HH
21251 7B        1400        ld   a,e
21252 2F        1410        cpl
21253 5F        1420        ld   e,a
21254 7A        1430        ld   a,d
21255 2F        1440        cpl
21256 57        1450        ld   d,a
21257 A7        1460 HH:    and  a
21258 ED52      1470        sbc  hl,de
21260 D1        1480        pop  de
21261 E1        1490        pop  hl
21262 3818      1500        jr   c,END
21264 42        1510        ld   b,d
21265 4B        1520        ld   c,e
21266 2A6E53    1530        ld   hl,(EEE)
21269 EB        1540        ex   de,hl
21270 2A7053    1550        ld   hl,(HHL)
21273 1B        1560        dec  de
21274 23        1570 END1:  inc  hl
21275 227053    1580        ld   (HHL),hl
21278 ED536E53  1590        ld   (EEE),de
21282 ED52      1600        sbc  hl,de
21284 DA3052    1610        jp   c,START
21287 C9        1620        ret
21288 44        1630 END:   ld   b,h
21289 4D        1640        ld   c,l
21290 2A6E53    1650        ld   hl,(EEE)
21293 EB        1660        ex   de,hl
21294 2A7053    1670        ld   hl,(HHL)
21297 18E7      1680        jr   END1
21299 7B        1690 PLOT:  ld   a,e
21300 E607      1700        and  7
21302 ED44      1710        neg
21304 C607      1720        add  a,7
21306 87        1730        add  a,a
21307 87        1740        add  a,a
21308 87        1750        add  a,a
21309 CB3A      1760        srl  d
21311 CB1B      1770        rr   e
21313 CB3A      1780        srl  d
21315 CB1B      1790        rr   e
21317 CB3A      1800        srl  d
21319 CB1B      1810        rr   e
21321 F6C0      1820        or   %11000110
21323 325353    1830 TV:    ld   (TT+1),a
21326 2A7253    1840        ld   hl,(HP)
21329 19        1850 TT:    add  hl,de
21330 CB46      1860 ZO:    bit  0,(hl)
21332 C9        1870        ret
21333 110080    1880 A_HL:  ld   de,49152
21336 4F        1890        ld   c,a
21337 E6F8      1900        and  248
21339 63        1910        ld   h,e
21340 6F        1920        ld   l,a
21341 29        1930        add  hl,hl
21342 EB        1940        ex   de,hl
21343 19        1950        add  hl,de
21344 EB        1960        ex   de,hl
21345 29        1970        add  hl,hl
21346 29        1980        add  hl,hl
21347 19        1990        add  hl,de
21348 B1        2000        or   c
21349 E607      2010        and  7
21351 87        2020        add  a,a
21352 87        2030        add  a,a
21353 87        2040        add  a,a
21354 84        2050        add  a,h
21355 67        2060        ld   h,a
21356 79        2070        ld   a,c
21357 C9        2080        ret
21358 0000      2090 EEE:   defw 0
21360 0000      2100 HHL:   defw 0
21362 0000      2110 HP:    defw 0
21364 0000      2120 XX:    defw 0
21366 0085      2130 YY:    defw 0

Pass 2 errors: 00

A_HL  5355   DD    5352   CIRCLE 5221
DE    52FF   EEE   536E   END    5328
END1  531A   HH    5309   HHL    5370
HP    5372   PLOT  5333   ST     5208
ST1   524E   START 5230   TT     5351
TV    5348   XX    5374   YY     5376

Table used:    182 from   383
Executes: 21000
```

či ugradnjom soft filtra) i ne sme se koristiti hardverski SCROLL. Ranije su u „Računarima" objavljeni skoro isto tako brzi softverski skrolovi, koje možete koristiti u svojim programima

Može li još brže? Da, naravno! Korišćenjem tabelarne A-HL rutine, ili potpuno novim algoritmima (za PLOT ili ček CIRCLE rutinu), kao što je, na primer, bejzik rutina iz primera 2 koja crta elipsu na potpuno nov način. Rutina na početku uzima konstante (početne vrednosti) SIN-usa i Cos-inusa, iz kojih kasnije izračunava ostala vrednosti potrebna za crtanje elipse. Rutina nije naročito brza jer je pisana u bejziku. Brzina joj se može povećati prevođenjem u mašinac nekim kompajlerom

Andrija Radović

„Atari ST"/„Fleet Street Publisher"

Štamparija na stolu

Davno najavljivan i dugo očekivan, na softverskom tržištu se konačno pojavio, pod imenom „Fleet Street Publisher", prvi program za stono izdavaštvo za računare „atari ST". Verzije koje posedujemo za ovaj prikaz (1,0 i 1,1), iako to nigde ne piše na disketi ili uputstvu za upotrebu, predstavljaju osakaćenu varijantu (tzv. level I) iz serije „Fleet Street Publisher" za IBM PC i „amstrad PCW", ali im ta pojednostavljenost ne oduzima previše na kvalitetu. Radi se o prvom upotrebljivom programu za stono izdavaštvo komercijalne klase, koji zaista zaslužuje ime koje nosi (Fleet Street — čuvena ulica štampe u Londonu.

Dzak Top Publishing ili „stono izdavaštvo uz pomoć RAČUNARA" predstavlja oblast računarstva koja se trenutno najbrže i najdramatičnije razvija. Razlog je verovatno, u tome što se danas 80% mikroračunara koristi samo za obradu teksta i što, gotovo svi mi, bili „profesionalni pisci" ili ne, želimo da naši tekstovi izgledaju „profesionalno" ma kakva da im je svrha. Kako kaže jedan francuski poslovni čovek (koji koristi „makintoš", Laserwriter i DTP softver) „Ako je vaša ideja dobra, onda će je kvalitetna prezentacija učiniti još boljom i prihvatljivijom".

Od koncepta do knjige

Sigurno je da softver za stono izdavaštvo (ako imate na raspolaganju odgovarajući računar i laserski štampač) omogućava da vaš tekst, izveštaj, reklamna poruka ili publikacija izgledaju kao da ste ih obratili profesionalnom štamparu. Na žalost stono izdavaštvo ni danas nije jeftino, jer su laserski štampači ona skupa stavka koja se ne može izbeći. Nada ipak ima, bar za aterijevce koji poseduju mašine za 2 i više MB, pošto će Atari uskoro izbaciti „glupi" a jeftini (1 300$) laserski printer koji koristi mikroprocesor i RAM u njihovim računarima da bi formirao stranicu A-4 formata sa rezolucijom od 12 tačaka po dužnom milimetru, što daje kvalitet otiska sličan ovome koji upravo čitate. Naravno, na ovim printerima je moguće otisnuti i grafiku (linijsku ili u sivim polutonovima), uključujući tu i fotografije u relativno grubom rasteru.

Sve zajedno, sedeći za svojim stolom, na svom kompjuteru možete napisati, izvršiti lekturu i korekturu, dizajnirati oblik i strukturu stranice i najzad odštampati (na svom laserskom printeru) u manjem broju primeraka — tekst, izveštaj, tehničku dokumentaciju, bilten, knjigu ili časopis. Odštampane stranice se mogu koristiti kao matrice za dalje umnožavanje na ofsetu ili nekoj drugoj štamparskoj tehnici.

Nije bez značaja i to da ovakav postupak omogućava znatno veću kreativnu kontrolu nad celokupnim procesom, kao i da ponekad dramatično, obara troškove budući da sami možete da obavite niz radova za koje biste inače morali da platite odgovarajuće stručnjake. Sistem za stono izdavaštvo vam neće postavljati ograničenja ako ste već dobar grafički dizajner, ali činjenica da njime raspolažete ne znači i da ćete to nužno i postati. Alat je, svakako važan deo



zanata, ali je za njegovo kreativno i produktivno korišćenje potrebno imati i mnogo stručnog znanja i posebne nadarenosti za određenu oblast. Nadajmo se da širenje stonog izdavaštva neće značiti i poplavu neinventivnog, dosadnog i ružnog grafičkog dizajna kada se njime počnu baviti ljudi koje toma privlači samo lakoća rada, funkcionalnost i jevtinoća postupka. Druga (lepša) strana ove stvari je da će se grafičkim dizajnom i izdavaštvom moći baviti i oni ljudi koji nemaju određena stručna

znanja (tipografije, na primer), jer će im programi za stono izdavaštvo omogućiti da sa lakoćom postižu „profesionalne rezultate" a da uopšte ne znaju šta je to „cicero", „špigl", „šlajfna", „punkt" i slično.

Mek je kriv za sve

Iako je elektronska obrada i priprema teksta već dugo u upotrebi, stono izdavaštvo kao posebna aplikacija na mikrokompjuterima nastaje pojavom ALDUS „Pagemaker"-a na „makintošu", kao i pojavom „Apple Laserwriter"-a. Kasnije se pojavilo mnoštvo softverskih paketa za stono izdavaštvo na MAC-u i IBM PC-u, pa i na „amstradu" i „komodoru". Treba reći da se još uvek najkvalitetniji programi za stono izdavaštvo „vrte" na MAC-u, kao i da je „Laserwriter" još uvek jedan od najkvalitetnijih manjih (stonih) laserskih printera.

Na najraširenijem personalnom kompjuteru danas, IBM PC-u, takođe se može koristiti nekoliko izvrsnih programa za stono izdavaštvo, kao što su već pomenuti „Pagemaker" kompanije Aldus i „Ventura" koju prodaje Xerox. Za rad sa ovim programima potrebno je, međutim, imati odgovarajuće grafičke karte, miše i Microsoft „Windows" ili Digital Research GEM nadgradnje MS DOS operativnog sistema.

Najjeftiniji upotrebljiv sistem za stono izdavaštvo možete dobiti nabavkom „atari ST" računara, odgovarajućeg programa (osim pomenutih FSE i FSP programa do sada se pojavio još i „Publishing Partner") i nekog jeftinijeg laserskog printera, kao što je OMS KS (ako ne želite da čekate Atarijev). Cena ovakvog sistema se u Engleskoj kreće oko 2 500 funti, dok su cene sličnih sistema sa MAC-om i PC-em nekoliko puta veće. Nije nevažno pomenuti da je i ovoga puta Atari demonstrirao da jeftino ne mora biti i loše, tj. da njihov reklamni slogan „za mase a ne za više klase" zaslužuje poverenje.

Priprema teksta

Program „Fleet Street Publisher" ima, u stvari, dva jasno odvojena dela: **tekst procesor** i **editor stranica**. Kada uđete u program, on vas „odmah uvodi" u tekst procesor. On je dosta dobrog kvaliteta, iako se, naravno, ne može nositi sa „1st Word"-om. Međutim, moguće je odmah preći u uređivač stranica, ako je vaš tekst već napisan, bilo u tekst procesoru FSP-a ili u nekom drugom tekst procesoru („1st Word", „Haba Word", itd.) kojima se može dobiti tekstualna datoteka u ASCII kodu. Ukoliko je, pak, vaša tekstualna datoteka u nekom od posebnih (tekst procesorskih) formata, vi je možete „spremiti" za korišćenje u FSP-u uz pomoć „Stripera", kratkog programa za „skidanje" posebnih kodova, tj. svođenja tekstualnih datoteka na ASCII kod.

Kad smo već kod pripremanja datoteka za korišćenje u FSP-u, treba reći i da u ovom programu možemo koristiti i ilustracije (grafičke datoteke — grafika iz piksel orijentisanih, tzv. „paint" grafičkih programa) iz drugih programa, bilo direktno iz „Graphic Director" (Mirrorsoft) programa, bilo indirektno, uz pomoć malog „Convert"



programa, koji formate grafičkih datoteka iz (dobro poznatih) „Gem Paint", „Degas" ili „Profi-Painter" programa prevodi u grafički format „Fleet Street Publisher"-a.

U svim „prozorima" (windows) FSP-a prati će vas niz „sličica" (icone) na levoj strani ekrana. Šest od njih se nalaze u grupi jedna iznad druge. To su pet sličica koje označavaju „modove": GRAFIČKI BLOK, GRAFIČKA INTERVENCIJA, ISCRTAVANJE OKVIRA I LINIJA, TEKSTUALNI BLOK (svi se odnose na EDITOR STRANICE) i tekst procesor. Peta sličica je indikator (kada pocrni) „viška teksta" koji je smešten u privremeni memorijski prostor. Dve poslednje, nezavisno postavljene sličice će sterijevci lako prepoznati. Prva je CLIPBOARD (tabla za crtanje i pisanje) koja označava privremenu memorijsku adresu za jednokratno odlaganje teksta ili grafike. Druga je kanta za smeće, gde može „baciti" (tj. obrisati) sve što nam ne treba.

Pored ovih pomagala, kao i uvek, na ST-u su nam na raspolaganju i meniji DESK, FILE, OPTIONS, TYPOGRAPHY i LAYOUT. Ovi se meniji mogu koristiti i u tekst procesoru i u editoru stranice, iako u prvom „ne rade" neke funkcije koje se odnose na dizajn stranica.

Prelom teksta

U drugi glavni mod ovog programa ulazi se tako što se iz OPTIONS menija izabere NEW PAGE funkcija. Kada učinimo ovaj izbor, pojaviće se dijaloški okvir (nosi naslov PAGE LAYOUT) koji nam omogućava da napravimo sledeći niz izbora: veličine papira (A-4, američki obični, legal, a mogu se definisati i drugi formati), broj stubaca na stranici, definisanje margine, širine stubaca, razmak između stubaca, itd. Kada se izvrši izbor, na ekranu će se pojaviti novi prozor koji će nositi obris stranice upravo onakav kakav smo izabrali. Linije naslovnog prostora i stubaca dati su isprekidanim crticama.

Ako želimo da u datu formu stranice „ulijemo" već napisani tekst (iz FSP tekst procesora ili „ogoljeni" ASCII fajl iz drugog tekst procesora, svejedno je) moramo prvo kliknuti mišem na funkciju TEKST BLOK-a, a zatim „ocrtati" blok na obrisu prvog stupca (ako tu želimo da smestimo tekst). Iscrtani (prazni ili puni) tekstualni blok se može premeštati ili kopirati „povlačenjem" miša i uključenjem SHIFT(za kopiranje) tastera. Veličina tekstualnog bloka može se, slično okvirima u grafičkim programima, menjati „potiskivanjem" ili „širenjem" (mišem, takođe) graničnih linija. Srazmerno smanjenje ili povećanje se postiže dijagonalnim potiskivanjem ili povlačenjem miša kada je strelica u samom uglu okvira. Ova tehnika rada sa okvirima je ista i kada se radi o grafičkim (praznim ili punim) blokovima.

Kada smo zadovoljni veličinom i razmeštajem stubaca (koji su sada ujedno i tekstualni blokovi), možemo „uliti" tekst u onaj u čijem gornjem levom uglu stoji kursor (što znači da je aktivan). To činimo birajući funkciju MERGE TEXT iz FILE menija. Na ovaj izbor pojavljuje se direktorijumski okvir (potpuno se koriste GEM rutine) iz kojeg, na uobičajeni način, biramo tekst koji smeštamo. Treba reći da tekst možemo i direktno da pišemo u napravljeni tekstualni blok, ali se tada ne možemo koristiti razvijenijim funkcijama tekst procesora, pa je ovu mogućnost bolje koristiti za prepravke teksta u zadnjem trenutku.

I elektronski slog

Kada je tekst „uliven" u formu stranice, s njim se može raditi sve (izbor tipografskih i dizajnerskih funkcija u ovom programu je zaista vrlo bogat). Moguće je najpre tekst „složiti" različitim fontovima (stilom slova) u raznim veličinama. Ovo se čini izborom odgovarajuće funkcije u TYPOGRAPHY meniju. Treba reći da se, za sada, nudi 12 tipova slova (sa polucrnim, **podvučeno**, kurzivnim, ili svetlim verzijama) u veličinama od 10 do 216 (ako imate 1 MB RAM-a) tačaka. Razmak između slova u reči, (kerning) razmak između reči u redu i razmak između redova u stupcu (leading) mogu se proizvoljno odrediti. Sve mere u FSP-u se mogu dati u mm, cm, pica, point apsolutnim i relativnim jedinicama, kao i procentima. Tekst se može „slagati" na različitim pozadinama (različite gustoće rastera), te belo na crno.

Tekst se, konačno, može staviti u ram (različitih debljina linija) ili se može otkucavati preko grafike. Najzad, blokovi teksta se mogu pomerati, kopirati, prebacivati, smeštati u privremenu memorijsku banku, itd. Osim pomenutog, još se mnogo toga može raditi sa tekstualnim blokovima (više tipova poravnavanja, više tipova deljenja reči, više načina tabuliranja, itd., itd.). Upotrebom TIDY LINKED TEXT funkcije iz LAYOUT menija postiže se „prelivanje" teksta iz jednog stubca u drugi u okviru jedne stranice. Ako je tekstualni fajl duži, onda onaj njegov deo koji ne može da stane na jednu stranicu odlazi u privremeno memo-







Poznata pop pevačica RIVA oduševljeno komentariše novi DTP program za računar "ATARI ST" - FLEET STREET PUBLISHER: je izvanredan paket jer mi omogućava da pravim novogodišnje čestitke rodbini" kaže ona.



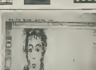
Sve na gotovo. Uz program „Fleet Street Publisher" isporučuje se i velika biblioteka crteža, slika i okvira

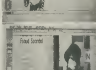
rijsko skladište (počev od šeste sličice s leve strane) pa kada „ispunimo" jednu stranu nju možemo posiati na disk a u predefinisanu formu stranica možemo smestiti nov komad teksta — i tako dok ne „ulijemo" ceo tekst

Ovde je prilika da učinimo jednu zamerku ovom, inače, sjajnom programu. On nije projektovan da automatski uliva tekst u niz stranica (kao što je to slučaj sa „Pagemakerom", „MS Word 3", „Venturom" i sličnim programima). To znači da se kod „ulivanja teksta" i štampanja završenih stranica mora malo više raditi tj. ponavljati neke operacije umesto da se jednostavno zada komanda koja bi se jednovremeno odnosila na niz stranica. Kad smo kod primedbi (ništa nije savršeno) našlo bi se svakako još. FSP, na primer, nema tzv. „picturewrap" mogućnost, tj. u njemu tekst ne može da „teče oko slika" — što je šteta jer se tako mogu dobiti lepi grafički efekti.

Likovno uređenje

Ilustracije (od tabela i grafikona preko tehničkih crteža i umetničkih ilustracija do fotografija) su, pored teksta, osnovni element svakog programa za stono izdavaštvo, pa se i u ovom softverskom paketu njima posvećuje dužna pažnja. Grafika koju prihvata ovaj program je piksel orijentisana (naziva se još i bit mapped) što pruža dosta velike mogućnosti za rad, ali nameće i određena ograničenja. Gotove ilustracije iz biblioteke koja se dobija na disku (kao i one koje sami napravite i prevedete u FSP format) lepe se u napravljenu formu stranica uz pomoć funkcije MERGE PICTURE iz FILE menija. Kao i u slučaju teksta pre toga je potrebno formirati blok za grafiku (kliknuti na sličicu na vrhu niza na levoj strani ekrana i „otvoriti" okvir odgovarajuće veličine) a onda, kada se slika „unese" na zadato mesto, možemo s njom raditi šta nam je volja. Osim što mogu





biti pomerane i kopirane na bilo koju poziciju na stranici, ilustracije se mogu smanjivati i povećavati i to proporcionalno (celina slika se ne menja, iako se one povećava ili smanjuje) i redukciono (slika gubi određeni deo, dok ostatak ostaje iste veličine) ilustracija se, takođe, može staviti u okvir čije linije mogu biti različitih debljina.

Što se tiče okvira i linija, pored jednostavnih, tj. ravnih i punih linija, mogu se koristiti i različiti kitnjasti „uzorci" koji se nalaze u grafičkoj biblioteci na posebnom disku.

Makro definicije

Na kraju treba još nešto reći o vrlo korisnoj funkciji KEY MACROS kojom se korišćenjem funkcijskih tipki (uz pomoć ALTERNATE tastera može se definisati 20 makro funkcija) mogu odrediti potpune tipografske karakteristike nekog tekstualnog bloka — pritiskom na jedan (plus ALTERNATE eventualno) taster. Kada definišete ove MAKRO naredbe, možete ih odložiti na disk za kasniju upotrebu. Moguć je gotovo beskonačan broj kombinacija MAKRO naredbi — jedino ograničenje je kapacitet diska. Tekst takođe može biti sačuvan kao KEY MACRO (heder futer ili bilo kakav drugi tekst koji želite da ponavljate na stranicama koje uređujete). Kada želite da ga ubacite na stranicu treba samo pritisnuti određeni funkcijski taster. Tekst se takođe može sačuvati kao grafička IMG (GEM grafički format) datoteka, pa sa s njim kao takvim može manipulirati kao sa nekom grafikom — što je posebno zgodno sa nekim naslovima koji se ponavljaju u različitim veličinama.

Jednu, zaista bitnu funkciju FSP, još nismo pomenuli — mogućnost pregleda cele stranice u različitim veličinama koje se izražavaju u procentima (original je dat kao 100%) u dijapazonu od 12% (stranice veličine nokta) do 200% (ekran prikazuje samo 1/6 stranice). Ova funkcija omogućava dobar uvid u dizajn stranice i lakšu manipulaciju njenim elementima.

Na kraju ostaje nam da još kažemo da je FSP na I nivou vrlo dobar program (jedini ozbiljan nedostatak je nepostojanje funkcije automatskog ulivanja teksta u niz stranica) dok bi FSP na II nivou morao biti još bolji budući da će, kako se obećava, uključiti i sledeće funkcije: automatsko ulivanje teksta u niz stranica, prihvatanje objekt orijentisane grafike, „izlivanje" viška teksta u unapred određenu stranicu, opcija pomoći na ekranu, rečnik za podelu reči na kraju reda, ugrađeni drajver za laserski printer (za I nivo se on posebno dokupljuje za 60 funti). Ali, kako saznajemo, ovaj program će zahtevati i posedovanje hard diska kao minimalnu konfiguraciju spoljne memorije.

„Fleet Street Publisher" staje u Londonu 110 funti, što je za nas u Jugoslaviji prilično skupo. Međutim, naši pirati vredno rade i već se jedna, istina okljaštrena, verzija ovog programa može nabaviti za deseti deo londonske cene (i to još u dinarima). Ukoliko imate „atari ST" i želite da pravite „svoje novine" ili da svom tekstu date „profesionalni izgled" — ovo je program za vas.

Milan Nikolić

IBM Professional Editor

Prava alatka za Prave Programere

Gotovo da nema programskog jezika za koji ne postoji verzija za IBM PC. U tom izobilju svako može pronaći nešto za sebe. Pravi raj za Pravog Programera! U sklopu gotovo svakog takvog paketa dolazi i neki editor za pisanje izvornog koda. Ti su editori, na žalost, i najslabija strana čitavog paketa. Tu su više reda radi: pružaju samo neke najosnovnije opcije, nekomforni su i neljubazni za korištenje . . . Sve u svemu, idealan put da se korisniku ogadi programiranje. Najbolje rješenje je korištenje nekog od specijalizovanih programa-editora. Specijalizovani editori, samim tim što su dizajnirani da budu oruđe programera, pružaju daleko veće mogućnosti i komfor u radu. Eliminiše se i potreba za savladavanjima raznih editora, jer taj editor koristimo za pisanje izvornog koda u bilo kom programskom jeziku. Pošto Pravi Programeri najviše vremena troše upravo na pisanje programa, odlučili smo da na ovim stranicama predstavimo jedan od najpopularnijih editora za IBM PC — IBM Professional Editor.

IBM Professional Editor (u daljem tekstu PE) je potpuni ekranski editor. Spada u stariju generaciju programa (nastao je davne 1982. godine). Potiče iz doba kada se i sam IBM brinuo o području svog, tada novog računara. Zato, kad god vidite IBM logo na početku programa, možete biti sigurni da je to neki „old-timer". Da je program pravljen relativno davno vidi se i po nekim drugim karakteristikama.

Jako je skroman u zahtjevima. Biće zadovoljan i sa jednim diskretnim pogonom, monohromnim grafičkim adapterom i 64 K RAM-a (verzija EDIT64). Za potpunu verziju programa biće ipak potrebno 96 K RAM-a. Ovako skromne zahtjeve lako je razumjeti ako se prisjetimo da je to, u stvari, bila osnovna konfiguracija PC-a vrijeme kada je počeo svoj pohod na tržište. Kao i PC, i PE će rado prihvatiti sva dalja proširenja sistema i pored ove skromnosti u zahtjevima, **Professional Editor** mogućnostima potpuno opravdava svoj naziv.

Ako još postoje neke nedoumice o tome šta se može raditi sa editorom, da ih razjasnimo.

Editor se može koristiti za pisanje izvornog koda u bilo kom programskom jeziku, pravljenje datoteka ulaznih podataka i kreiranje tekstualnih dokumenata. Program jednostavno pozivamo sa diskete ili diska navođenjem njegovog imena. Ukoliko se neko nije igrao komandom rename, to bi trebalo da bude EDIT (EDIT 64 za skraćenu verziju koja radi na minimalnoj konfiguraciji). Za trenutak će se pojaviti IBM logo a onda prelazimo u editor, tj. u njegov inicijalni meni.

Profilna datoteka

Prije početka rada potrebno je navesti imena datoteka sa kojima ćemo raditi. Ukoliko tek počinjemo rad, unijećemo samo ime (sa nastavkom) izlazne datoteke. Polje sa imenom „misteriozne" profilne datoteke (PROFILE FILE) automatski se popunjava. Ime profilne datoteke sastavljeno je od nastavka (extension) izlazne datoteke i nastavka PRF. U toj datoteci čuvaju se informacije o parametrima editora za određeni tip ulazne (izlazne) datoteke. Ako, na primjer, pišemo program u paskalu (nastavak PAS), profilna datoteka će imati ime PAS.PRF. Ako smo već ranije radili sa datotekama tog tipa, tj. ako se PAS.PRF nalazi na radnom disku, koristiće se ta datoteka za postavljanje parametara editora. Ukoliko nema takve datoteke, otvoriće se nova sa standardnim (default) parametrima. Parametri se mogu postaviti u toku inicijalizacije, iz menija koji se pozivaju funkcijskim tipkama.

Profile Options Menu (F4) daje mogućnost postavljanja osnovnih parametara za rad sa editorom.

Tabs&Mask Options Menu (F5) koristi se za postavljanje tabulatora i maski za unos linija.

Pomoću Data Macro definitions Menu (F6) možemo dodijeliti svakoj od alfanumeričkih tipki frazu koja će se pojaviti u tekstu svaki put kada tu tipku pritisnemo u kombinaciji sa Alt tipkom. Na raspolaganju su dva tipa makroa. Za kraće sekvence (do 30 karaktera) koristi se P (Phrase) tip. Sekvenca se definiše u DATA polju. Za duže sekvence, koje mogu imati i više linija, koristi se F (File) tip. U DATA polje se unosi ime datoteke koja sadrži odgovarajuću sekvencu. Tip makroa mora se obavezno definisati u polju T.

Prelaskom u editor (F9) svi ovi parametri se zapisuju u profilnu datoteku. Promjene parametara u toku rada nema efekta na profilnu datoteku.

Ekran je podijeljen na tri dijela. Prvih osam kolona popunjenih tačkicama rezervisano je za linijske komande. Preostalih 140 (stotinu četrdeset) kolona koriste se za unos teksta. Tekst zauzima 24 linije ekrana dok posljednja, dvadeset peta, služi za prikaz aktivnog seta funkcijskih tipki i statusa. Dvadeset četvrta linija je komandna linija. Služi za unos editorskih komandi i prikaz statusa. Kada nije u upotrebi služi za prikaz teksta.

Kretanje po tekstu

Kurzorski dio tastature, pored uobičajene funkcije kretanja po tekstu, ima i nekoliko drugih funkcija. Razlikujemo:

1. Normal mode
2. Control mode
3. Scroll mode
4. Numeric mode i
5. Data mode

Koji je način trenutno aktivan zavisi od toga koja je od tipki Num Lock, Scroll Lock, Ctrl ili Alt pritisnuta u kombinaciji sa tipkama sa kurzorskog dijela.

1. Normal mode

Num Lock i Scroll Lock su isključeni. Druge tipke nisu pritisnute. Za kretanje po tekstu najčešće ćemo koristiti ovaj način.

— Kurzorske strelice su zadržale svoju standardnu funkciju.
— Home postavlja kurzor na početak linije.
— End briše sve od kurzora do kraja linije.
— PgUp i PgDn pomjeraju tekst za vrijednost vertikalnog pomjeraja (vertical scroll increment) gore, odnosno dole. Standardna vrijednost je 12 linija.

2. Control mode

Dobija se iz normalnog načina kada držimo pritisnutu Ctrl tipku. Korištenjem ovog načina ostvaruju se specijalne funkcije.

— Lijeva i desna kurzorska strelica pomjeraju kurzor za čitavu riječ lijevo ili desno. Druge dvije strelice nemaju funkciju.
— Home postavlja kurzor na kraj linije.
— End obnavlja prethodnu vrijednost polja u meniju ili linije u tekstu.
— PgUp i PgDn postavljaju kurzor na početak, tj. kraj teksta.

3. Scroll mode

Iz normalnog načina se u ovaj način prelazi kada se pritisne Scroll Lock tipka. Koristi se za brže pomjeranje po tekstu.

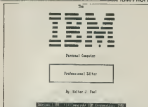
— Kurzorske strelice pomjeraju tekst na ekranu za zadani broj linija ili kolona.
— Home pamti trenutnu poziciju kurzora za kasnije korištenje.
— End pamti trenutnu poziciju kurzora i kurzor prebacuje na poziciju sačuvanu prethodnim korištenjem Home tipke i nazad.
— PgUp i PgDn postavljaju kurzor na početak, tj. kraj teksta.

4. Numeric mode

U ovaj način se prelazi kada je pritisnuta **Num Lock** tipka. Koristi se kao numerička tastatura kod unosa veće količine podataka.

5. Data mode

U ovaj način se dolazi iz normalnog kada se drži pritisnuta Alt tipka. Koristi se za unos specijalnih znakova iz proširenog seta karaktera



Pritisnute tipke se tretiraju kao decimalna vrijednost ASCII koda karaktera. Ako na primjer želimo na ekranu dobiti karakter čiji je ASCII kod 188 pritisnućemo **Alt** tipku na kurzorskom dijelu ćemo otkucati redom 1 8 8 i na kraju pustiti Alt tipku. Karakter sa tim kodom pojaviće se na ekranu na mjestu kurzora. Ovom tehnikom se na primjer, korištenjem blok grafike mogu uokviriti dijelovi teksta ili postići neki drugi efekti.

Tipke **Ins**, **Del** i **Back Space** imaju svoju uobičajenu funkciju. Važno je napomenuti da se zadnjih 140 obrisanih karaktera čuvaju na posebnom steku i mogu se obnoviti pritiskom na tipku **Esc** — po jednom za svaki karakter.

Osnovne funkcije

Pored kurzorskog dijela višestruku funkciju imaju i funkcijske tipke. Na raspolaganju je 28 funkcija koje su podijeljene u 4 seta. Koji je set trenutno aktivan pokazuje broj u krajnjem lijevom donjem uglu ekrana. Neke funkcije su na raspolaganju tek ako su prethodno zadovoljeni uslovi za njihovu primjenu.

Set 0 — osnovni set

F01 (F11, F21, F31) — KEYS

Služi za izbor odgovarajućeg seta. Promjena je ciklična.

F02 — SWAP

Prebacuje kurzor sa tekuće pozicije u komandnu liniju (linija 24) i nazad.

F03 — LINE

Prebacuje kurzor sa tekuće pozicije u oblast rezervisanu za linijske komande i nazad.

F04 — RECALL

Sa komandnog steka vraća prethodno unešenu komandu.

F05 — INSERT

Prebacuje **PE** u mod za unos (input mode) i ubacuje praznu liniju iza tekuće linije.

F06 — DELETE

Briše tekuću liniju sa ekrana i iz datoteke. Ukoliko je linija obrisana greškom, može se vratiti **UNDELE** funkcijskom tipkom.

F07 — CANCEL

Opoziv linijske komande.

F08

— Ne koristi se

F09 — SRCH D

Izvršava **SEARCH** komandu od pozicije kurzora prema kraju teksta (Downward).

F10 — CHNG D

Izvršava **CHANGE** komandu od pozicije kurzora prema kraju teksta.

Set 1 — kraj rada, brisanje i dijeljenje linije i uključi/isključi funkcije

F12

— Ne koristi se

F13 — END

Završetak rada sa **PE**. Prethodno snima tekst na disk.

F14 — QUIT

Prekid rada sa **PE**. Ne formira se izlazna datoteka.

F15 — WRAP

Uključuje ili isključuje automatski prenos dijela riječi koji prelazi desnu marginu u novi red.

F16 — UNDELE

Omogućava obnavljanje linije pogrešno obrisane **DELETE** funkcijskom tipkom (**F06**). Moguće je obnoviti posljednjih pet obrisanih linija.

F17 — SPLIT

Od pozicije kurzora liniju dijeli na dvije linije. Moguća samo u edit modu.

F18 — MASK

Uključuje ili isključuje masku za unos.

F19 — SCROLL

Uključuje ili isključuje automatsko pomjeranje slike na ekranu (scrolling) kada kurzor dođe na jednu od njegovih ivica.

F20 — INDENT

Uključuje ili isključuje automatsko uvlačenje linija.

Set 2 — Pretraga i zamjena u svim pravcima

F22

— Ne koristi se

F23 — SRCH U

Izvršava **SEARCH** komandu od tekuće pozicije kurzora prema početku teksta (Upward).

F24 — SRCH F

Izvršava **SEARCH** komandu od početka teksta tražeći prvu (First) pojavu uzorka.

F25 — SRCH D

F26 — SRCH L

Izvršava **SEARCH** komandu od kraja teksta tražeći zadnju (Last) pojavu uzorka.

F27 — CHNG U

Izvršava **CHANGE** komandu od tekuće pozicije kurzora prema početku teksta.

F28 — CHNG F

Izvršava **CHANGE** komandu od početka teksta.

F29 — CHNG D

F30 — CHNG L

Izvršava **CHANGE** komandu od kraja teksta.

Set 3 — kontrola štampača

F32 — PRINT

Sve linije koje su trenutno na ekranu šalje na printer. Štampa se samo tekst, bez statusne linije i specijalnih poruka.

F33 — SPACE

Prazna linija na štampaču.

F34 — EJECT

Štampač prelazi na početak nove stranice.
Ostale funkcijske tipke se ne koriste.

Linijske komande

Linijske komande namijenjene su manipulaciji sa blokovima linija. Zadaju se u za to posebno izdvojenoj oblasti uz lijevu marginu. Sve imaju sličnu sintaksu, lako se pamte, elegantne su i efikasne.

Linijske komande sastoje se od jedne ili više potkomandi koje definišu vrstu operacije, početak i kraj bloka i eventualno odredište.

Da bismo zadali linijsku komandu potrebno je pritisnuti **F03** tipku koja kurzor iz tekuće linije prebacuje u oblast rezervisanu za linijske komande.

Neke komande za izvršenje zahtijevaju unos potkomandi na raznim linijama. Kada se sve potkomande unesu, komanda se izvršava a potkomande se brišu iz komandne oblasti. Potkomande se ne moraju unositi na istom ekranu niti je važan redoslijed unosa.

Kada se unese prva potkomanda, omogućuje se funkcija **CANCEL** (**F07**) kojom se mogu poništiti sve potkomande nekompletne linijske komande. Pojedinačne potkomande mogu se poništiti tako da se preko njih otkucaju blank karakteri.

Prije nego se linijska komanda kompletira moguće je vratiti se u tekst i izvršiti neke izmjene ili zadati neku od editorskih komandi. Ukoliko obrišemo liniju na kojoj se potkomanda odnosila komanda se automatski briše.

Linijska komanda se može sastojati od jedne ili više potkomandi koje se zadaju u proizvoljnom redoslijedu velikim ili malim slovima, u oblasti rezervisanoj za linijske komande

Neke komande imaju opcionalni brojčani operand koji se unosi odmah iza komande se ili bez razmaka između To je cijeli broj u rasponu između 1 i 65535 Negativni brojevi označavaju se znakom minus (−) Znak zvjezdice (*) koristi se da označi sve linije (* predstavlja „beskonačnu" vrijednost od 65535) Ako broj nije naveden podrazumijeva se 1

Potkomande označavaju vrstu operacije i granicu bloka linija Prilikom unosa mogu se koristiti dva formata

U prvom formatu potkomanda se predstavlja kao

X n

X (1 karakter) definiše o kojoj se operaciji radi, a n predstavlja brojčani operator Ukoliko je zadan pozitivan broj operacija se odnosi na tekuću i n-1 narednih linija Ako je broj negativan operacija se odnosi na tekuću i n-1 prethodnih linija Ako je kao n unašena *, blok sadrži tekuću i sve linije do kraja teksta Ako je unešena vrijednost -*, obuhvaćena su tekuća i sve linije do početka

U drugom formatu sub-komanda se unosi kao

XX

XX (2 karaktera) ponovo predstavlja operaciju ali istovremeno i početak ili kraj bloka Da bi se blok ograničio potrebno je unijeti obe granice

Linijske komande kojima se kopiraju ili pomjeraju blokovi linija zahtijevaju da bude definisano i odredište Potkomande A i B određuju mjesto na koje će se umetnuti blok Ukoliko se neka od ovih potkomandi unese unutar označenog bloka javlja se greška i komanda se ne izvršava

Preostaje nam još samo da se upoznamo sa raspoloživim linijskim potkomandama Slijede samo navedene njihove funkcije dok za sintaksu vrijedi sve što je navedeno ispred U zagradama su navedeni oblici za prvi i drugi format respektivno

After (A)

Potkomanda odredišta. Označava da će se blok umetnuti iza linije na kojoj je zadana ova potkomanda

Before (B)

Potkomanda odredišta Označava da će blok biti umetnut ispred linije na kojoj je zadana ova potkomanda

Copy (C n ili CC)

Kopira jednu ili više linija na poziciju određenu potkomandama odredišta.

Copy and repeat (CR n ili CCR)

Identična je običnoj **Copy** naredbi s tom razlikom da se nakon izvršenja ne briše potkomanda za definisanje bloka linija već samo odredišna potkomanda da bi se moglo zadati novo odredište Komanda se mora okončati **CANCEL** funkcijskom tipkom **(F07).**

Delete (D n ili DD)

Briše jednu ili više linija Linije obrisane na ovaj način ne mogu se obnoviti

Extract (E n ili EE)

Jednu ili više linija kopira u posebnu datoteku prethodno otvorenu editorskom komandom **EOPEN** Ta datoteka ostaje otvorena sve dok se ne zatvori editorskom komandom **ECLOSE** Komanda **END** ili **QUIT** automatski zatvaraju i ovu datoteku

Gather (G n ili GG)

Nekoliko linija komprimirati u jednu Uslov je da rezultujuća linija ne bude duža od 140 karaktera

Move (M n ili MM)

Premješta jednu ili više linija na poziciju određenu odredišnim potkomandama

Print (P n ili PP)

Štampa jednu ili više, linija na štampaču Štampaju se samo linije koje sadrže tekst.

Uppercase/Lowercase (U n ili UU/L n ili LL)

Prevodi mala slova u velika odnosno velika slova u mala.

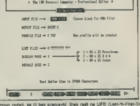
Column grid (COL)

Koristi se za prikaz linije sa oznakama kolona. Ta linija se pojavljuje ispred linije na koju se komanda odnosi Ova linija se ne upisuje u datoteku niti se pojavljuje na štampaču Po potrebi se može obrisati korištenjem **DELETE** funkcije **(F06).**

Editorske komande

Pored linijskih postoje i posebne editorske komande U komandni mod prelazimo pritiskom na **SWAP** funkcijsku tipku **(F02). PE** u ovom modu ostaje sve dok ponovo ne pritisnemo **F02**

PE raspolaže posebnim komandnim stekom na kom se čuvaju zadnje četiri komande To može uštediti nešto vremena kod višestrukog zadavanja komande jer se u svakom trenutku komanda može pozvati sa steka i



ponovo zadati, sa ili bez prepravki Stek radi na LIFO (Last-In-First-Out) principu a komande se poziva **RECALL** (funkcijskom tipkom **(F04).** Jednim pritiskom poziva se komanda sa prvog (najvišeg) nivoa drugim sa drugog itd Poslije četvrtog nivoa ponovo se vraćamo na prvi Nakon izvršenja komande se vraća na isti nivo sa koga je i pozvana.

Sintaksa ovih komandi malo je komplikovanija. Usvojimo zato jedinstveno predstavljanje

Komanda se može zadati velikim ili malim slovima punim nazivom ili skraćenicom Znak označava ili-ili opciju (jedno ili drugo) Parametri koji su napisani malim slovima su obavezni Parametri u malim zagradama su opcionalni

BOTTOM B

Skok na kraj teksta

BOUNDS BDU lbound OFF rbound

Postavlja lijevu **(lbound)** i desnu **(rbound)** kolonu, koje se koriste kao granice za **SEARCH** i **CHANGE** komande Lijeva granica mora biti manja ili jednaka desnoj Dozvoljene vrijednosti su od 1 do 140 **OFF** parametar se koristi za reset granica na njihove standardne vrijednosti 1 i 140

CHANGE C /string1/string2/ (n) (lbound (rbound))

Ovom komandom se u tekstu prvih n pojava jednog uzorka **(string1)** zamjenjuju drugim uzorkom **(string2).** Ukoliko su zadane granice traženja i zamjene vršiće se samo za uzorke koji se potpuno nalaze unutar granica Mjesto prvog uzorka ne može biti prazno Maksimalna dužina uzoraka su 40 karaktera Dozvoljene vrijednosti za n su 0-65535 Može se koristiti i * za oznaku „beskonačnog" ponavljanja. Ako nije naveden operator n, podrazumijeva se 0 Time su definisani uzorci ali nikakva zamjena se ne vrši Ovim se međutim omogućavaju funkcijske tipke **SRCH X** i **CHNG X** (X predstavlja neki od mogućih pravaca pretrage) kojima se sprovodi selektivna zamjena Pretraga u željenom pravcu započinje pritiskom na odgovarajuću **SRCH** funkcijsku tipku Ukoliko smo zadovoljni zamjenu vršimo pritiskom na **CHNG D** tipku U protivnom nastavljamo pretragu

Granice zadane u sklopu komande imaju prioritet u odnosu na one zadane komandom **BOUNDS.**

Postoje i varijacije **CHANGE** komande u odnosu na poziciju od koje počinje pretraga

CHANGED CD
CHANGEU CU
CHANGEF CF
CHANGEL CL

Kada su jednom zadani argumenti komande funkcijskim tipkama ili navođenjem neke od ovih varijacija komande bez parametara može se ponoviti operacija zamjene

COLUMN COL

Isto što i **COL** linijska komanda

DOWN D (n)

Pomjeranje po tekstu za trenutnu vrijednost vertikalnog pomjeraja (vertical scroll increment) ili za n linija nadole

ECLOSE EC

Zatvara datoteku prethodno otvorenu komandom **EOPEN**

END E

Učinak ekvivalentan korištenju **END** funkcijske tipke

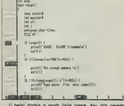
EOPEN EO filename

Otvara datoteku u koju će se smještati dijelovi teksta izdvojeni **EXTRACT** linijskom komandom

FILES F

Prikazuje imena trenutno aktivnih datoteka i njihov status

GET G (n)*



U bafer dodaje n novih linija teksta. Ako nije naveden broj linija podrazumijeva se * i u bafer se smješta maksimalni broj linija. Ako u baferu već postoji tekst, novi tekst se dodaje na kraj.

LEFT:L (n)

Pomjeranje po tekstu za vrijednost horizontalnog pomjeraja (horizontal scroll increment) ili za zadani broj linija, lijevo

LINEND

Vraća broj tekuće linije u odnosu na prvu liniju u tekst baferu

MACRO:MAC X T (text)

Naknadna definicija ili promjena makroa. X predstavlja tipku, T je tip, a text definicija makroa

MARGIN MAR leftmargin: =:OFF rightmargin: =:OFF

Postavlja, ili mijenja, margine u toku rada. Dozvoljene vrijednosti su 0—132 za lijevu i 0 ili 5—137 za desnu marginu. Desna margina mora biti bar za 5 veća od lijeve. Karakter = može se koristiti da se sadrži prethodna vrijednost margine. OFF opcija dezaktivira marginu (daje joj nultu vrijednost)

MASK MA

Definisanje ili promjena maske za unos linija

MERGE M filename

Dodaje tekst iz neke druge datoteke tekstu sa kojim trenutno radimo. Tekst se dodaje iza tekuće linije. Ako sav novi tekst ne može stati u bafer, prijavi se greška i novi tekst se ne dodaje

NEXT N (n:*)

Ako se radi sa velikim tekstom koji se obrađuje u segmentima, ovom komandom se sav tekst ili samo njegovih prvih n linija zapisuje u izlaznu datoteku, a isto toliko sa novog teksta učitava u tekst bafer. Ako nije naveden broj, podrazumijevaju se linije od početka do tekuće linije

PUT P (n:*)

Zapisuje navedeni broj linija u izlaznu datoteku. Ako broj nije naveden zapisuju se sve linije od početka do tekuće linije

QUIT:Q

Učinak ekvivalentan korištenju QUIT funkcijske tipke

RIGHT R (n)

SEARCH:S /string/ (n:*) (lbound (rbound))

Za ovu komandu važe sva razmatranja kao i za komandu **CHANGE**

SET

Koristi se za promjenu vrijednosti raznih editorskih opcija u toku rada sa PE. Neke od varijanti neće biti detaljnije obrađene jer su jasne same po sebi

SET ASCII EXTENDED:STANDARD

SET AUDIO 1:2:3

1-zvučni signal samo kod greške
2-samo poruke i greške
3-sve potvrde i poruke

SET RESERVE n

Mijenja veličinu bafera, n je veličina bafera u Kb i može varirati između 2 i 63 K. Efekti promjene vidljivi su nakon korištenja **GET** komande

SET DISK A:B:C

SET HSCROLL HS n N:P

Postavlja veličinu horizontalnog pomjeraja, n može biti u rasponu od 0—72. N označava pola stranice (40 kolona) a P punu stranicu (72 kolone)

SET PARA n

Uvlačenje paragrafa za n kolona, n je u rasponu između 0 i 60

SET LPI 6:8

Broj linija po inču na štampaču

SET WIDTH 80:132

Širina stranice na štampaču (normalni ili kondenzovani ispis)

SET UNIT 0:1:2

Trenutno aktivni štampač

SET SCREEN M80:B80:C80

SET VSCROLL VS n H P

TABS TA

TOP:T

UP U (n)

Nakon ovog podužeg spiska instrukcija treba nešto reći i o načinu na koji PE radi sa datotekama. Sve datoteke se čitaju i upisuju u standardnom DOS formatu. Ulazna datoteka je otvorena samo u trenutku čitanja. U toku rada PE kreira privremenu datoteku za izlaz. U slučaju normalnog završetka rada (END) originalna ulazna datoteka se briše, a izlazna datoteka biva sačuvana pod imenom ulazne datoteke. Ako je izabrana **BACKUP** opcija, originalna datoteka dobija nastavak **BAK** i ne briše se.

U slučaju nenormalnog prekida rada ulazna datoteka ostaje nedirnuta.

Zbog kreiranja privremene datoteke mora se paziti da na radnom disku ima dovoljno prostora za još jednu kopiju ulazne datoteke i eventualne dodatke. U slučaju da na disku više nema slobodnog prostora sistem će zahtijevati da se specificira drugi disk na koji će se moći smjestiti ostatak datoteke.

Ako je prostor na disku problem, može se koristiti opcija direktne promjene ulazne datoteke (**edit in place**). Privremene datoteke se ne formira i sve promjene se direktno upisuju u originalnu datoteku koja je otvorena u toku čitavog rada.

PE tekst bafer kreira od memorije preostale nakon učitavanja programa, ali najviše do veličine od 64 K. To ograničenje ne važi i za datoteke koje se obrađuju pomoću PE. Datoteke, koje su prevelike da stanu u bafer procesiraju se u segmentima

Formatiranje teksta

Iako je primarno namijenjen pisanju izvornog koda PE raspolaže i sa nekim osnovnim funkcijama za formatiranje pa se može koristiti i za pisanja kraćih tekstova.

Blok linija teksta može biti formatiran tako da bude poravnata lijeva margina a da riječi ne prelaze desnu marginu. Može se poravnati i desna margina ili tekst centrirati između margina. Za sve operacije koristi se slijedećih nekoliko linijskih komandi

Format (F n ili FF)

Poravnava tekst po lijevoj margini. Riječi koje prelaze desnu marginu prebacuje u novu liniju. Prvu liniju bloka (paragrafa) uvlači

Justify (J n ili JJ)

Poravnava tekst po obe margine i uvlači prvu liniju

Center (K n ili KK)

Centrira blok linija između margina

Tabs (T n ili TT)

Uklanja tabulatore iz označenog bloka

Ovim bi bila gotova naša priča o PE. Ako puno programirate ovo je prava stvar za vas. Program spada u „klasiku" i vrijedi ga nabaviti. Njegove osobine kao što su

— potpuni ekranski red
— dužina linije od 140 karaktera (zgodno za „nazubljene" programe),
— korištenje makroa, pogotovo File tipa, i maski za unos
— jednostavno i elegantno manipulisanje blokovima linija
— moćna komanda za pretraživanje i zamjenu,
— čuvanje obrisanih zadnjih 5 linija i 140 karaktera
— korištenje proširenog ASCII seta
— mogućnost obrade dužih datoteka u segmentima,
— čuvanje integriteta ulazne datoteke
— automatsko pravljenje bakap kopije (po želji) i
— mali zahtjevi na hardver

čine ga zaista moćnim i korisnim programerskim alatom

Na žalost, nedostatak **help** opcije sa spisom instrukcija (da ne spominjem uputstvo za upotrebu) učinio je da ovaj program na našim geografskim širinama nedoživi veliku popularnost. Ovaj tekst je, nadam se, uspio da bar donekle ispravi tu nepravdu

Naravno ni PE nije savršen. Ako bi neko bio sitničav našle bi se poneka mana.

Poželjno bi bilo radnu datoteku automatski u kraćim razmacima spašavati na disk. Rad sa više datoteka istovremeno, jednostavan prelaz iz jedne u drugu i mogućnost prikaza bar dvije datoteke na ekranu u posebnim prozorima bile bi, takođe, vrlo korisne opcije. Komande bi se mogle birati iz menija i trebalo bi ih biti manje ali ne na račun efikasnosti itd

No to su sve osobine novijih programskih proizvoda. Neki od njih će možda biti prikazani na ovim stranicama, ali o tom potom

Do tada uživajte u radu sa **Professional Editor-om**

Arif Agović

Crtanje na bejziku (3)

S lopticom u planine

Ništa nije nedostižno za našu lopticu, pa ni odlazak u planine. Pre nego što nastavimo njene doživljaje, podsetimo se na to da je čovek merilo svih vrednosti i pogledajmo šta se dešava sa čovečuljkom iz prethodnog broja.

U programu vidimo dva petlje čiji je zadatak da oživa čovečuljka. Prva od njih se koristi da bi se čovečuljek pomereo duž neke putanje i ta petlja se naziva glavne animacijske petlje. Unutar nje se nalazi druga petlja koja služi za izmenu slika (faza) koje čine ciklus hode. To je, ujedno, najprostiji vid povezivanje kretanje like i kretanja unutar lika.

Izuzeci bez pravila

Pravilno smenjivanje faza je cilj kome moramo težiti u kreiranju naših likova, ali svako poželi bar nekad da napravi nešto neuobičajeno čime će zadiviti gledaoce. Tako u mnogim slučajevima kriterijum za smenjivanje faza neće biti tako očigleden, pa će nekada biti potrebno da se formira novi niz podataka, koji govori o tome kada se koja faza crta.

Na slici 1 su predstavljene tri faze u kretanju loptice. Pretpostavili smo da se loptica izdužuje kad se kreće naviše, a da se sažima kad pada. Između ta dva deformacije, postoji prelazni period u kome loptica nije izobličena. Ovo, svakako, ne mora da ima neke naročite veze sa stvarnošću, pa i navedene deformacije ne moraju da odgovaraju stvarnom stanju stvari, ali je važno da se postigne odgovarajući vizuelni efekat. Sami možete da isprobavate razne kriterijume za izmene faza nekog lika pa i da „ručno" formirate niz upisujući član po član.

Sada za animaciju jedne kratke sekvence kristimo tri niza u kojima su smešteni četiri podaci. Nizovi X() i Y() čuvaju koordinate iscrtavanje lika, a niz A() sadrži brojeve slika koje se crtaju. Ako imamo u vidu da se za spremanje svakog člana niza koristi pet bajtova („spektrum" nema dvobajtne cele brojeve) i da želimo u proseku 150—300 slika za jednu kraću sekvencu, vidimo da apetit za memorijom naglo raste. Navedena sekvenca, na primer, u ovakvom obliku zahteva oko 1,5 kilobajta. Zato treba ozbiljno razmisliti o pakovanju podataka radi uštede memorije.

Sličice u paketu

U prošlom broju smo vam sugerisali jedan način u kome se konsultuju sistemske promenljiva koje sadrže koordinate poslednje nacrtane tačke. Ovaj metod je zahtevao prepravljanje nekih od navedenih rutina.

Isti zadatak se može obaviti i pomoću posebne procedura koje će tri niza X(), Y() A() smestiti u jedan znakovni niz čije su dimenzije (3, n), gde je n dimenzija bilo kog od prethodne tri niza. To je moguće jer ni jedna od (stvarnih) koordinata „spektrumovog" ekrana ne prelazi vrednost 255, što je i maksimalan sadržaj jednobajtne informacije. Kako ne verujemo da ćete praviti više od 255 faza za neki lik, tako ni niz A() ne bi trebalo da pravi probleme. Ako sve navedene podatke smestimo, na primer u niz Z$ (3, n), tada će pri iscrtavanju desete slike biti korišteni sledeći podaci: Z$ (1, 10) za x koordinatu, Z$ (2, 10) za y koordinatu i Z$ (3, 10) za broj slika koja će se iscrtati. Naravno da za pretvaranje znake u broj treba upotrebiti standardnu funkciju CODE.

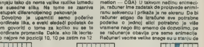
Pomenućemo još jedan način pakovanje podataka, efikasniji od prethodnog. Sama animacija ima smisla ako se koordinate lika menjaju postepeno. Nema efekta ako lik nacrtamo na koordinatama 10, 10 i zatim ga naglo preselimo na poziciju 150, 150. Efekat kretanje se potpuno gubi. Zato bi crtanje išlo otprilika ovim redom: 10, 10, 12, 12, 14, 14... Vidimo da se koordinate menjaju tako da nema velike razlike između dve susedne slika. Na tome se zasniva postupak diferencijalnog pakovanje.

Dovoljno je upamtiti samo početne koordinate lika, a svaki sledeći podatak će nam govoriti o tome za koliko su se te koordinate promenile. Dakle, ako lik iscrtamo najpre na poziciji 10, 10 pa zatim na 12 12 onda će prvi podatak iz niza glasiti 2,2 što je razlika između koordinate dvaju slika, po čemu je i ovaj način pakovanje dobio ime. Ako se radi o nekoj sporo promenljivoj veličini, kao što su koordinate animiranog lika, onda je ovaj način vrlo pogodan. Za smeštanje razlike x koordinata koristimo tri bita, za y razliku takođe tri bita (što nam daje slobodu od ±4 tačke). Ako za smeštanje faze uzmemo dva bita (što dopušta četiri slike za jedan lik), ukupni podaci za iscrtavanje jedne slika se mogu smestiti u samo jedan bajt.

Pakovanje obavite sami u zavisnosti od vaših potreba. Ako pravite crtani film, biće vam svakako potrebno, ali za manje sekvence i eksperimentisanje nije neophodno.

Levo, desno, nigde mog ekrana

Ako ste se pitali, čemu služe sva ova spremanje i pakovanje podataka evo i odgovore. Da bi se dobile računarski proizvedene animacije (Computer Generated Animation — CGA). U takvom načinu animacije, računar ima zadatak da proizvede animiranu sekvencu i prikaže je na ekranu. Da bi računar stigao da izračuna sve potrebne podatke o jednoj slici potrebno je više vremena nego što smemo da dozvolimo, pa se računanje obavlja pre same animacije. Računari veoma velike snage su u stanju da

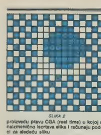


1 2 3

SLIKA 1

```
 10 REM      RACUNARI 28
    — s lopticom u planine —
    —       V  Basic 1987   —

 20 REM trofazna loptica
 30 REM       48Klk
 40
 50 DIM a$(3,9)
 60 RESTORE 760
 70 FOR j=1 TO 3
 80   FOR i=1 TO 9
 90     READ a
100     LET a$(j,i)=CHR$ a
110   NEXT i
120 NEXT j
130
140
150 REM putanja (trajektorija)
160
170 LET f$="30*EXP (-t/10)*ABS
    (COS(t))"
180 LET min=0,max=10*PI,frames=
    144
190 fmaxp f$,min,max,frames
200 PRINT #0;"pritisni taster"
210 PAUSE 0
220 CLS
230
240
250 REM sledeci deo programa
260 REM kaju fazu (sliku) lopta
270 REM crtamo na ekranu
280 REM : A() broj slike
290
300 DIM a(frames)
310 LET a(1)=2
320 FOR i=2 TO frames
330   LET a=y(i)-y(i-1),a=a*(AB
      S a)>0)
340   LET a(i)=2+SGN a
350 NEXT i
360
370
380 REM pasiv procedura za
390 REM animaciju; crta sa
400 REM slika a$(1,2 i 3)
410 REM x(),y() koordinate
420 REM a() broj slika
430
440 f animate x(),y(),a(),2
450 PRINT #0;"Jos jednom?"
460 PAUSE 0
470 IF INKEY$<>"d" THEN GO TO 4
    80
480 CLS
490 GO TO 440
500
510
520 REM smestujemo nizove
530 REM x(),y() i a() na jednie
540 REM znakovnih nizova a$(3,n)
550
560 pack a(),y(),a(),z$
570 z animate a$,2
580 PRINT #0;"Jos jednom?"
590 PAUSE 0
600 IF INKEY$<>"d" THEN GO TO 6
    00
610 CLS
620 GO TO 570
630
640
650 REM sa pozadinom
660 CLS
670 pozadina
680 f_animate a(),y(),a(),2
690 PRINT #0;"Jos jednom?"
700 PAUSE 0
710 IF INKEY$<>"d" THEN GO TO 8
    00
720 CLS
730 GO TO 680
740
750
760 REM loptica 1,2,3
770 DATA 0,0,24,126,255,126,24
    0,0
780 DATA 0,0,56,124,124,124,56,
    0,0
790 DATA 0,16,56,56,124,124,56,
    56,16
800
810
820 REM primer iscrtavanja sa
830 REM prioritetom
840 REM m$ maske lika
850 REM a$ uzorci lika
860 REM x(),y() koordinate
870 REM a() broj slike
880
890 pozadina
900 sp animate a(),y(),a(),2
910
920 STOP
930
940
950
```



SLIKA 2

proizvedu pravu CGA (real time) u kojoj se naizmenično iscrtava slika i računaju podeci za sledeću sliku

Svi ovi problemi otpadaju ako želite samo da radite uz pomoć računara (Computer Aided Animation) Pri tome nije kritično vreme rada računara i smeštanje podataka jer se slika sa ekrana snima trik kamerom (frame by frame) i kasnije reprodukuje normalnom brzinom

```
7200>REM RACUNARI 28
     ---- animacija sa fazama ----

7201
7202 DEF PROC f_animate REF x(),
     REF y(), REF a(),n
       DEFAULT n=1
       LOCAL i
       PLOT OVER 1;x(1),y(1);a$(
       a(1))
       FOR i=2 TO LENGTH(1,"x(")
         PLOT OVER 1;x(i-1),y(i-
         1);a$(a(i-1))
         PLOT OVER 1;x(i),y(i);a
         $(a(i))
         PAUSE n
       NEXT i
     END PROC
7203
```

Loptica nikad ne dolazi sama

Tu je i pozadina Loptica se nalazi u nekom ambijentu, pa pogledajmo šta stvara probleme u tom slučaju

Najprostiji način da se iscrta loptica je da se definisani lik jednostavno spusti u video memoriju Ovakav čin bi imao dve posledice

1 Lik je definisan u obliku pravougaonika ili kvadrata (8x8 8x16 16x16 tačkica) pa bi loptica na šahovskoj tabli izgledala kao na slici 2

2 Sadržaj koji se do tada nalazio u video memoriji na tom mestu je bespovratno izgubljen

Poslednji problem se može rešiti tako što se sadržaj video memorije privremeno prebaci na neko drugo mesto Naravno problem postoji samo kod računara koji nemaju sprajtova

Pravi problem se rešava tako što se video čip opremi različitim načinima crtanja To su u stvari, mogućnosti da se između lika i sadržaja ekrana izvrše razne logička operacije

Postoji li način da se objedini rešenje ova dva problema Ako ste primetili da pri crtanju stalno koristimo atribut OVER 1 i upitali se za razlog, onda ste na pravom putu za iznalaženje rešenja

Atribut OVER saopštava način preklapanja novog i starog sadržaja ekrana Za vrednost 0 efekat će biti kao na slici 2 OVER 1 je signal da se između starog i novog sadržaja izvrši operacija ISKLJUČIVO ILI (XOR ili EOR) Beta bejzik dodaje još i mogućnost upotrebe običnog ili (OR) što se opisuje atributom OVER 2 Tu je još i atribut INVERSE koji može da dâ negativ lika koji se crta (operacija NOT)

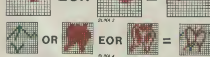
Za nas je najinteresantnija operacija EOR jer se posle dvostruka primena ove operacije sa istim sadržajem vraća prvobitni rezultat Pojasnimo to izrazom

A EOR B=C C EOR B=A

Kako ovo izgleda u praksi, možete videti na slici 3 Očigledno da više nije potrebno čuvati pozadinu, već se nakon dva iste crtanja pozadina u potpunosti restaurira. Postupak je brz i efikasan ako vam ne smeta to što likovi liče na duhove Na ovaj način radi i većina komercijalnih programa (igara) u kojima je važnija brzina nego lepota crtanja



SLIKA 3



SLIKA 4

```
7300 REM RACUNARI 28
     ---- preklapanje ----

7301
7302 DEF PROC pup x,y
       GET p$ x,y-2,2
     END PROC
7303
7304 DEF PROC pdown x,y
       PLOT x,y;p$
     END PROC
7305
7306 DEF PROC pat a$,x,y
       PLOT x y;a$
     END PROC
7307
7308 DEF PROC mask a$ x,y
       PLOT x,y,a$
     END PROC
7309
7310>DEF PROC pp_animate x(),y()
     ,z(),z
       DEFAULT n=1
       LOCAL i
       pup x(1),y(1)
       mask a$(z(1)),x,y
       pat z$(z(1)),x,y
       FOR i=2 TO LENGTH(1,"x(")
         pdown x,y
         pup x(i-1),y(i-1)
         mask a$(z(i)),x,y
         pat s$(z(i)),x y
         PAUSE n
       NEXT i
     END PROC
7400

7401>DEF PROC pozadina
7402 DEF PROC pozadina
       LOCAL i$ $
       KEYWORDS $ 784
       FOR i=1 TO 784
         PRINT CHR$ 134;
       NEXT i
       KEYWORDS 2
     END PROC
7403       :

8300>REM RACUNARI 28
     ---- pakovanje podataka ----

8301
8302 DEF PROC pack REF x(), REF
     y(), REF a(), REF z$
       LOCAL i
       DIM z$(3,LENGTH(1 "x("))
       FOR i=1 TO LENGTH(1, x(")
         LET z$(1 i)=CHR$ (256*( 8305
         x(i)+xsm)/xrg)
         LET z$(2,i)=CHR$ ((75*(
         y(i)+ysm)/yrg)
         LET z$(3,i)=CHR$ z(i)
       NEXT i
       DELETE x()
       DELETE y()
       DELETE z()
     END PROC
8303
8304>DEF PROC z_animate REF z$,z
       DEFAULT n=1
       LOCAL i
       LET xrg=256,yrg=174,xsm=0
       ,ysm=0
       PLOT OVER 1;CODE z$(1,1),
       CODE z$(2,1);a$(CODE z$(3
       ,1))
       FOR i=2 TO LENGTH(2,"z$("
       )
         PLOT OVER 1;CODE z$(1,i
         -1),CODE z$(2,i-1);a$(C
         ODE z$(3,i-1))
         PLOT OVER 1;CODE z$(1,i
         ),CODE z$(2,i);a$(CODE
         z$(3,i))
         PAUSE n
       NEXT i
     END PROC
```

U nekoliko nivoa

Tu je najzad i način crtanje koji je prikazan na slici 4 Pri ovakvom načinu potrebno je definisati sliku u više nivoa Prvi nivo zove se maska i služi tome da „očisti" izvesno područje ekrana Drugi nivo naziva se uzorak (pattern) i iscrtava sliku na pripremljenoj podlozi (mask) Uzorak more biti u negativu, to jest ono što je na uzorku „osvetljeno" na ekranu će biti „ugašeno"

Postoje naravno i drugi načini iscrtavanja slike zavisno od potrebe i načina definicije i od raspoloživih modaliteta crtanje Pogledajmo izraz

back AND NOT mask OR pattern

i videćemo da je ovde potreban pozitivan uzorak i negativna maska

Svaki od likova može da ima svoj prioritet koji ukazuje na redosled iscrtavanja, pri čemu pozadina ima najniži prioritet Kod poslednje dva načina je potrebno sačuvati deo ekranske memorije na kome se crta, radi kasnije restauracije

Poslednji deo programa koji koristi upravo ovakvo iscrtavanje je više demonstracionog karaktera, jer bejzik nije dovoljno brz za ovakav način crtanje

Efekat koji se postiže prikazan je na slici 4, na kojoj se vidi kako se postiže „oreol" oko lika koji bolje razdvaja lik od pozadine. Da bi stvar bila jasnija lik i pozadina su dati u različitim bojama

Računarski algoritmi (5)

Metodi eliminacije

„Metodima eliminacije" nastavlja se serije „Računarski algoritmi" u kojoj prof. dr Dušan Slavić daje niz algoritama sa programima za rešavanje odabranih numeričkih problema. U ovom tekstu reč je o rešavanju sistema linearnih jednačina sa proizvoljnim brojem nepoznatih. To je novi primer za tvrđenje da se i teški numerički problemi mogu pretočiti u jednostavan algoritam i rezumljiv program.

Povod za pisanja ovog nastavka serija „Računarski algoritmi" je višestruk. Pre svega to je potreba za jednostavnim i efikasnim programom za rešavanje sistema linearnih jednačina. Taj problem je „uvek aktuelan". To je jedan od odlučnih primera protiv da se matematička nauka ne razvija baš tako brzo kako nam se ponekad čini. Najzad, treba odlučnije ustati protiv vrlo rasprostranjenog (a pri tome i vrlo štetnog) gledišta da je za pisanje računarskog programa dovoljno uzeti knjigu iz matematike ili numeričke analize i skicu postupka (da ne kažem algoritam) prenati na bejzik ili neki drugi programski jezik. Ovde je priložen jedan takav program iz domaće literature. Naravno, dat je i korekten program.

Apstraktni matematičari

Dat je sistem od n linearnih (algebarskih) jednačina sa n nepoznatih

$$\Sigma_{j=1,n}\ a(i,j)\times(j)=b(i) \qquad i=1(1)n$$

Oznaka i=j(k)n znači da „i" uzima vrednosti od j u koraku po k do n. Koeficijenti a(i,j) i b(i) su zadani a vrednosti x(j) su nepoznate.

U matričnom obliku ovaj sistem može se jednostavno zapisati

$$AX=B$$

gde je A matrica sistema dimenzija n*n, X vektor nepoznatih (jednokolona matrice dimenzija n*1) i B je vektor slobodnih članove (takođe jednokolona matrica dimenzija n*1). Za apstraktnog matematičare problem je „rešen". Istorija nauka uči da nema rešenih problema, već samo problema manje ili više rešavenih. „Rešavanja" sistema za apstraktnog matematičara sastoji se u množenju matrične jednačine sleve sa A^{-1} (inverznom matricom matrice A)

$$A^{-1}AX=A^{-1}B$$

zatim je $A^{-1}A$ jednako jediničnoj matrici I (dimenzija nxn)

$$IX=A^{-1}B$$

i proizvod jedinična matrice I sa X je X, pa je

$$X=A^{-1}B$$

Jednostavno, lepo, elegantno — ali ima dve male mane: nalaženja inverzne matrice je oko tri puta komplikovanija od rešavanja sistema linearnih jednačine i napisati A^{-1} ne znači umeti izračunati vrednost A^{-1}. Operacija sa matricama biće obrađena u jednom od nastavaka ove serija. Inverznom matricom celishodno je rešavati sistem ako je matrica sistema nepromenjena bar za tri razne vrednosti vektora slobodnih članova.

Kod apstraktnih matematičara na ceni je takozvani Kramerov metod za koji je znao (mnogo pre Kramera) Isak Njutn. Rasprostranjena je zabluda u udžbenicima numeričke analize (ne svim) da broj operacija množenja potrebnih ovom metodu raste sa faktorijalom broja jednačina n. To je tačno samo ako se vrednosti determinanata matrice računaju razvijanjem po elementima vrste ili kolone (po uobičajanoj definiciji) — dakle na najgori mogući način. Ako se determinante računaju Gausovim algoritmom, onda je broj operacija množenja za sistem srazmeran četvrtom stepenu od n, što je nesumnjivo manje loše nego da je srazmeran faktorijelu od n. Može se reći da Kramerov metod ima „veliki teorijski značaj" (što treba shvatiti ovako: praktičan značaj tog metoda je skoro nikakav).

Šalu na stranu, treba se ipak okrenuti metodima kod kojih je potreban broj operacija množenja srazmeran trećem stepenu broja jednačina n. Iz rečenog nikako ne sledi da Kramerov metod treba odbaciti. Na primer, ako su koeficijenti regularnog sistema celi brojevi, onda su rešenja racionalni brojevi — pa metod koji se Krameru pripisuje omogućava da se nepoznate odrede tačnije, a posebno određuju vrednosti brojilaca i vrednost imenilaca nepoznatih, te vrednosti su celi brojevi u smislu matematike ali ne i računarstva.

Povratna zamena

Jedan slučaj sistema linearnih jednačina je veoma značajan pa ga treba posebno razmatrati. Neka svi koeficijenti ispod dijagonale sistema imaju vrednost nulu

$$a(i,j)=0 \quad (i>j)$$

Tada je mogućno sistem linearnih jednačina jednostavno rešiti. Iz poslednje jednačina izračuna se vrednost poslednje nepoznate x(n), pa se zamenom vrednosti x(n) u pretposlednjoj jednačini izračuna vrednost pretposlednje nepoznate x(n−1), zamenom vrednosti x(n) i x(n−1) u jednačini ispred ove dve dobije se x(n−2) i tako dalje dok se zamenom vrednosti x(n), x(n−1), ..., x(2) u prvoj jednačini ne dobija vrednost prve promenljive k(1). Ovaj postupak obično se naziva „povratna zamena". Postoji jedna izmena ovog postupka. Nađena vrednost x(n) iz poslednje jednačina može se uvrstiti u sve prethodne jednačine, čime se menjaju slobodni članovi ovih jednačina i od sistema reda n dobija sistem reda n−1.

Zatim se iz poslednje jednačina tog novog sistema izračuna vrednost x(n−1) i on uvrsti u sve prethodne jednačine (ima ih n−2 na broju) i tako dalje, dok se uvrštavanjem vrednosti x(2) u jedinu preostalu jednačinu ne izračuna vrednost x(1). Broj operacija u oba metoda je isti. Ipak ne treba dati brzopletu ocenu da su ove dva metoda računarski ekvivalentna. U matematici važi asocijativnost sabiranja (a+b)+c=a+(b+c), pa je to možda tamo i dopušteno, ali u računarstvu ne važi asocijativnost sabiranja, pa treba biti ili preterano hrabar ili nepromišljen pa očekivati da će oba metoda dati do poslednjeg bita isti rezultat. Ovom posebnom slučaju sis-

```
0   '       Računari 28 (jul 1987)
10          Dušan Slavić, Izmena Gaussovog metoda eliminacija nepoznatih
20          Treba uneti  N, broj nepoznatih sistema linearnih jednačina.
30  INPUT N:DEFDBL A-H,O-Z: NN=N*N: NN=NN+N: DIM A(NN)
40          Treba uneti  koeficijente proširene matrice sistema po vrstama.
50  FOR K=1 TO N: FOR I=K TO NN STEP N: INPUT A(I): NEXT I: NEXT K
60          Ispisivanje vrednosti  koeficijenata  u cilju provere učitavanja.
70  FOR K=1 TO N: FOR I=K TO NN STEP N: PRINT A(I);:NEXT I: PRINT: NEXT K
100         Gaussov metod eliminacije nepoznatih sa delimičnom pivotizacijom.
110 L=-N: N1=N+1: FOR J=1 TO N: L=L+N1: B=0: M=L
120         Traženje elementa kolone  matrice  sistema  maksimalnog  modula.
130 FOR K=L TO J*N: A=ABS(A(K)): IF A>B THEN B=A: M=K
140         Ako je matrica sistema bliska singularnoj  -  zaustaviti program.
150 NEXT K: B=A(M): IF B=0 THEN PRINT "sistem nije regularan": STOP
160         Eventualna zamena delova  vrsta  proširene  matrice  sistema  A.
170 IF M>L THEN B=-B:FOR K=L TO NN STEP N:A=A(M):A(M)=A(K):A(K)=A:M=M+N:NEXT
180         Eliminisanje promenljive u delu kolone  ispod  glavne dijagonale.
190 FOR I=1 TO N-J: A=-A(L+I)/B
200         Izmena proširene matrice sistema  od vodećeg člana  dole i desno.
210 FOR M=L+I+N TO NN STEP N: A(M)=A(M-I)*A+A(M): NEXT M: NEXT I: NEXT J
220         Nepoznate se računaju  obrnutim redom  -  od poslednje  ka prvoj.
230 L=NN: FOR I=N TO 1 STEP -1: A=-A(NN+I): M=L: B=-A(L): L=L-N1
240         Računanje se vrši  povratnom  zamenom  (gornji trougaoni sistem).
250 FOR K=I+1 TO N: M=M+N: A=A(M)*A(K)+A: NEXT K: A(I)=A/B: NEXT I
260 '       Rezultati  su u prvih  N  članova  proširene  matrice  sistema  A.
270 FOR J=1 TO N: PRINT "X(";J;") = ";A(J): NEXT J: END
```

tema linearnih jednačina daje s razlogom ovolikí prostor, jer se u opštem slučaju sistem linearnih jednačina pokušava da svede na ovaj slučaj. Kažem „pokušava" jer taj posao nije bez stranputice.

Gausov metod

Gaus (K. F. Gauss) je početkom prošlog veka dao metod koji proizvoljni sistem svodi na ovaj poseban slučaj, pa onda koristi povratnu zamenu.

Kod Gausovog metoda eliminacije nepoznatih broj operacija množenja je $n(n-1)(2n+5)/6$, ili grubo $n^3/3$. To je oko n puta brže nego kod Cramerovog metoda. Bilo je mnogo pokušaja da se ovaj broj operacija smanji, ali u tome se nije uspelo, jer od Gausova doba nije prošlo ni dva veka. Ko je rekao da se nauka razvija eksponencijalnom brzinom? Nauka se veoma sporo razvija. Razvoj nauke ne treba ocenjivati po broju radova, već po stvarnom razvoju postojećih metoda. Nove (bolje) metode su prava retkost.

Posmatrajmo matricu sistema A i vektor slobodnih članova B kao jednu celinu koja se obično naziva proširena matrica sistema dimenzije $n^*(n+1)$. Neka su zadani koeficijenti i slobodni članovi označeni indeksom 1

$$\Sigma_{j=1,n}\, a_1(i,j)\, x(j)=a_1(i,n+1) \quad i=1(1)n,$$

gde je vektor slobodnih članova B postao $n+1$-va kolona matrice A.

Ideja Gausovog algoritma je da se pomoću prve jednačina eliminiše prva nepoznata x(1) u jednačinama od druge do poslednje, pri čemu se koeficijenti tih jednačina menjaju. Zatim se pomoću druge (izmenjena) jednačine eliminiše druga nepoznata x(2) u jednačinama od treće do poslednje (koje se ponovo menjaju) i tako dalje, dok se ne dobije gornji trougaoni sistem (svi koeficijenti ispod glavne dijagonale su nule). Zatim se koristi povratna zamena.

Ovaj Gausov postupak mnogi autori su pokušali da prevaziđu. Želelo se da se otkrije efikasniji postupak koji bi se manjim brojem operacija množenja i sabiranja mogao da direktno da rešenje sistema linearnih jednačina. U tome se nije uspelo, već su se samo za neke posebne slučajeve našli efikasniji metodi, npr. kod „retkih" matrica (imaju veliki broj koeficijenata sa vrednošću 0) i kod simetričnih matrica.

Bukvalni prevod

Brojni autori numeričkih analiza veliku pažnju posvećuju Gausovom metodu. Savremena numerička analiza se teško može zamisliti bez Gausovog algoritma za rešavanje sistema linearnih jednačina eliminacijom nepoznatih. U novijim knjigama mogu se naći i programi koji realizuju Gausov algoritam na nekom od programskih jezika, ali to rade prilično traljavo. Izostavljam imena autora — jer to je primer kako ne treba pisati program, bar ne dok zagovornici „veštačke inteligencije" ne ispune svoja velika obećanja da će računari postati inteligentni i da će prepravljati ono što programeri rade u ono što bi programeri želeli da rade.

Sledeći primer biće dovoljan da pokaže ograničenost ovakvih programa. Sistem jednačina

$$0x+y=4 \quad x+y=7$$

ima rešenje $x=3$, $y=4$, ali ovakav program daje $x=0$, $y=1$. Uzrok pogrešnog rezultata je bukvalna primena Gausovog algoritma: očigledno nije dovoljno pronaći u literaturi skicu nekog postupka i prevesti je na bejzik. Deoba nulom nije dozvoljena, a upravo ona dovodi do pogrešnog rezultata, jer je $a(1,1)=0$. Neki autori tvrde da se upotrebom pivotizacije „dobijaju bolji rezultati". Reč „bolji" treba izostaviti jer se bez pivotizacije (kao što primer pokazuje) ne dobijaju rezultati.

Spasonosna pivotizacija

Delimična pivotizacija je izbor elementa najveće apsolutne vrednosti u jednoj koloni (od dijagonalnog elementa do poslednjeg elementa njegove kolone). Potpuna pivotizacija je izbor elementa najveće apsolutne vrednosti u matrici sistema od dijagonalnog elementa do svih elemenata ispod i desno od njega. Cilj pivotizacije je da se pomoću izmene dve vrste proširene matrice sistema svedu na najmanju meru numeričke greške izračunavanja i izbegne deljenje nulom. To je izmena Gausovog izvornog algoritma koja se pokazala neophodnom. Teško je konstruisati primer koji bi pokazao nedovoljnost delimične (parcijalne) pivotizacije i potrebu za potpunom (totalnom) pivotizacijom. Pre eliminacije promenljive od nekog dijagonalnog elementa do kraja kolone treba utvrditi koji element ima najveću apsolutnu vrednost i permutovati vrstu tog elementa sa vrstom dijagonalnog elementa. Kako su elementi tih vrsta levo od posmatrane kolone jednaki nuli, dovoljno je permutovati samo elemente na preostalim delovima vrsta.

Za $k=1$ (1) $n-1$ treba izračunati

$$\max_{i=k(1)n} \text{abs}\,(a_k(i,k))=\text{abs}(a_k(1_k,k))$$

pa ako je $a_k(1_k,k)=0$ prekinuti izračunavanje ($d=0$), inače ako je $1_k>k$ zameniti delove vrsta

$$a_k(k,j) \text{ i } a_k(1_k,j) \quad (j=k\ (1)\ n+1)$$

a zatim za $j=k+1$ (1) $n+1$ izračunati

$$a_{k+1}(k,j)=a_k(k,j)/a_k(k,k),$$
$$a_{k+1}(i,j)=a_k(i,j)-a_{k+1}(k,j)\, a_k(i,k) \quad (i=k+1\ (1)\ n)$$

Ako je $a_n(n,n)=0$ prekinuti izračunavanje ($d=0$), inače je rešenje

$$x_n=a_n(n,n+1)/a_n(n,n)$$
$$x_i=a_{i+1}(i,n+1)-\Sigma_{j=i+1,n}\, a_{i+1}(i,j)\, x(j) \quad (i=n-1\ (-1)\ 1),$$

a vrednost determinante je

$$d=(-1)^w \quad \Pi_{i=1,n}\, a_i(i,i)$$

gde je w broj izvršenih permutacija vrsta. Vrednost determinante d svakako treba objaviti, jer je ona indikator da li je sistem regularan.

Pokazalo se da moguće znatno ubrzati Gausov algoritam sa delimičnom pivotizacijom — ako se proširena matrica sistema posmatra kao vektor dimenzije $n^*(n+1)$ koji se dobija kada se kolone proširene matrice sistema nastave jedna na drugu, kao i ako se broj indeksiranja svede na najmanju moguću meru.

Neki autori pivotizacije izvode tako što upoređuju apsolutnu vrednost nekog elementa ispod glavne dijagonale sa apsolutnom vrednošću elementa na glavnoj dijagonali i ako je veće zamenjuju te vrste. To je nepromišljeno, jer očigledno imaju mnogo više permutacija nego što je to neophodno, imaju mnogo premeštanja vrednosti elemenata matrice sa mesta na kojem ne treba da budu — na mesto na kojem još manje treba da budu. Za permutaciju vrsta treba da bude kriterijum naći ne element veće apsolutne vrednosti — već naći element najveće apsolutne vrednosti. Tako se ima najviše $n-1$ permutacija. Trajanje indeksiranja za permutacije uopšte nije zanemarljivo, pa u prikazivanju efikasnosti algoritma ne treba posmatrati samo broj operacija množenja koeficijenata matrice, tim pre što mnogi računari izvode operacije sa celim brojevima pomoću operacija sa realnim brojevima. Čuvani program SIMQ iz SSP (paket naučnih programa firme IBM) ima znatno više deljenja koja su nepotrebna, ali njihov autor je verovao da je svejedno da li se deli zbir ili svaki sabirak.

Neprevaziđen algoritam

Jordan je nekoliko decenija posle Gausa dao metod eliminacije koji se obično naziva „Gaus-Jordanov metod" ili „Jordanov metod". Pominjanje Gausa je nekorektno jer ovaj metod ima broj operacija množenja reda $n^3/2$, što znači da ima oko 50 posto više množenja. Reći da je Jordanov metod varijanta Gausovog metoda je takođe neumesno iako ima sličnosti među tim metodima. Jordanov metod eliminacije sadrži ideju da se u svakoj fazi ne anuliraju samo elementi na koloni ispod dijagonalnog elementa već i iznad dijagonalnog. To znači da se ne dobija gornji trougaoni sistem (kao kod Gausa) već dijagonalni sistem. Kod Jordanovog metoda eliminacija je dvostruko duža nego kod Gausovog metoda, ali nema povratne zamene. I Jordanov metod može biti sa delimičnom ili potpunom pivotizacijom.

Čitav vek posle Jordana a vek i po posle Gausa Crout je dao metod koji neki autori smatraju boljim od Gausovog metoda. Drugi autori tvrde da su Croutov i Gausov metod ekvivalentni. Razlika između tih metoda je u redosledu operacija. Sa gledišta matematičke analize i numeričke analize (zbog asocijativnosti sabiranja) može se reći da je Croutov metod nova interpretacija Gausovog metoda. Oni koji su oduševljeni dekompozicijama matrica reći će da je Croutov metod logički elegantniji od svog znatno starijeg brata blizanca. Rečeno uopšte ne znači da su ta dva metoda računarski ekvivalentna (kako neki autori tvrde), jer se zbog neizbežnih grešaka u elementarnim računskim operacijama obično ne dobija identičan rezultat; u računarstvu može se reći da su to ekvivalentni metodi samo uz mnogo dobre volje. Uostalom, ako se tvrdi da je novi metod samo prerušen stari metod — onda nam Croutov metod ne treba kad već imamo Gausov.

Dušan Slavić

Klub Z80

Da li je nula?

Programer bez dovoljno iskustva neće mnogo razmišljati kako da utvrdi da li je vrednost promenljive na nekoj adresi nula. Jednostavno, napuniće akumulator tom vrednošću i primeniti naredbu CP (compare). Ukoliko je pri tom potrebno umanjiti vrednost promenljive, najverovatnije će napisati

```
LD A, (VAR)
CP 0
CALL Z, ADDR
DEC A
LD (VAR) A
```

Ove operacije zauzimaju 10 bajtova i zajedno traju 47 taktova. Elegantnije rešenje ima jedan bajt i tri takta manje

```
LD A, (VAR)
OR A
CALL Z, ADDR
DEC A
LD (VAR), A
```

isti rezultat ali sa dva bajta i sedam taktova manje može se postići na ovaj način

```
LD A, (VAR)
DEC A
CALL Z ADDR
LD (VAR), A
```

Još bolje rešenje postiže se ako negde na početku programa registarskom paru HL ili IX dodeli adresa promenljive (Na primer LD HL VAR) Kasnije je u programu dovoljno napisati

```
DEC (HL)
CALL Z, ADDR
```

Svega dve naredbe, četiri bajta i 21 takt Znači, efekat isti kao u prvom primeru a značajno ubrzanje uz uštedu u memoriji

Za optimizaciju programa u ovom smislu naročito je pogodan 16-bitni indeksni registar IX. U dobro uređenom programu sve promenljive su obično skupljene na jednom mestu Na primer

```
VAR0 DEFS 1
VAR1 DEFS 1
VAR2 DEFS 1
```

i tako dalje Negde na početku programa treba napisati

```
LD IX, VAR0
```

Time se postiže da IX ukazuje na VAR0, IX+1 na VAR1, IX+2 na VAR2 Ako treba, na primer, da sadržaj VAR16 posle dekrementiranja testiramo na nulu napisaćemo

```
DEC (IX+16)
CALL Z, ADDR
```

Naravno, ovo testiranje je moguće i posle suprotne operacije, to jest inkrementiranja

```
INC (IX+16)
CALL Z, ADDR
```

Napomena da će promenljiva postati nula ako je pre inkrementiranja imala vrednost 255 verovatno nije potrebna ali nije loše da se podsetimo No šta da se radi ako nije potrebno ni inkrementiranje ni dekrementiranje? Poznato je da naredba LD ne utiče na flegove Znači moramo se vratiti na početak

```
LD A, (IX+2)
CP 0
CALL Z, ADDR
```

Ne moramo! Logička disjunkcija akumulatora sa samim sobom neće mu promeniti vrednost, ali će setovati ziro ukoliko je A=0. Prema tome, imamo rešenje s jednim bajtom i tri takta manje

```
LD A, (IX+2)
OR A
CALL Z, ADDR
```

Može i ovako

```
XOR A
OR (IX+2)
CALL Z ADDR
```

Broj taktova je isti kao u prethodnom primeru ali je ušteđen jedan bajt Ovo rešenje ima još jedan potpuni ekvivalent

```
SUB A
OR (IX+2)
CALL Z, ADDR
```

Jedna nula, a toliko pisanja! reći će neko. Ali taj neko verovatno nije član Kluba Z80 pa mu ne padaju na pamet ni nova pitanja

Šta ako promenljivoj nije dodeljena adresa nego na njenu vrednost ukazuje neki registar recimo H ili E?

A šta ako je promenljiva veće od 255 to jest ako operišemo sa 16-bitnim brojem?

Utešimo se odgovorom da ništa nije konačno Na te nove pitanje odgovorićemo drugom prilikom a možda će neki član kluba ponuditi još bolje rešenje za probleme obrađene u ovom broju

Umalo da zaboravimo sigurno znate da IX (kao i IY) može da ukazuje na 255 relativnih adresa, od IX—128 do IX+127

LDIR PREPORUČENO

Vladimir Janković (Hilen[illegible] 5 Beograd) jedan je od desetak čitalaca koji su tražili da objasnimo upotrebu GENS-a i MONS-a. Na žalost prostor kluba je još mali za sada predlažemo listanje starih brojeva „Računara" i drugih kompjuterskih [illegible] u kojima se svojevremeno [illegible] pisalo o ovom paketu [illegible].

[illegible] Matić (I L Ribara 38, [illegible] Miloševac) predlaže da u [illegible] uvrstimo i amstradovce ko[illegible] mašinskom nivou takođe [illegible] Z80'. **Antonio Pezo** (Iva [illegible] 1, Dubrovnik) mišljenja [illegible] mogao da ima [illegible] posebne sekcije. Prve dve [illegible] okupljene oko mikroprocesora Z80 („spektrum" i [illegible]) a druge dve sačinjavali [illegible] (6502) i [illegible]. Najviše predlaga je [illegible] ima tri sekcije, [illegible] za Z80, spektrumovci [illegible] vlasnici [illegible] redakciji „Računara" [illegible] predlozima se [illegible].

[illegible] Sergio Čerškov (Dalma[illegible], Kaštel Gomilica) [illegible] izračunavanje kvadratnog [illegible] iz 16-bitnih celih brojeva [illegible]. Uskoro [illegible]

Razbarušeni sprajtovi



uređuje Vlada Stojiljković

LOAD „ja“

QUO VADIS, D.V. ili: Kako školu opremiti računarom

Vi ste D V — učenik usmerenjak, vitez bez mane i straha tvrd u naumu da zatrete sve Scile i Haribde koja su se isprečila plemenitim namerama da naša sitna dečica, u školskim klupama, dobiju računar

Igre ima tri nivoa ili faze U prvoj fazi pomažete svom nastavniku da otkrije gde je u školi sakriven računar (Naime brižni direktor Pantelija već je odavno nabavio računar ali shvata da su u tu mašinu uložena znatna društvena sredstva, pa je neda neodgovornoj dečuriji da je namilosrdno arče Nasuprot njemu nastavnik Joce zagrejan je za opštu stvar — ali nije toliko mlad i naivan da se tek tako suprodstavi direktoru) Lokacija se određuju na RND, (zavisno od škola) a mogu biti

— saksije u predvorju, ovo je u principu korisno — ako je prostorija sunčana a zalivanje redovno, čipovi mogu proklijati i narasti (od osmobitnih postanu šesnaestobitni),

— konj sa hvataljkama u fiskulutrnoj sali (dabome, o trojanskom konju zna se sve osim jedne sitnice će ii sa rita);

— odžak na krovu, poznato je koliko se redovno čiste,

— redni sto sekretara škola, ovde pazite da vam materijali o reformi obrazovanja na pritisnu prsta,

— školski sef, u principu, ovde ste nemoćni ako sa obližnjeg ćoška ne pozovete lokalnog sliedžiju, koji se poodavno specijalizovao za takvu vrstu poseta školi

Ako ste pronašli računar, uiazite u drugu fazu Potrebno je da neko predaje informatiku, tj da se računar zaista i koristi u školi Tu Joce otkriva svoje muke.

1 Mala mu je plata

2 Ima ženu dvoje dece taštu sa malom penzijom, a sva njih treba izdržavati

3 Shodno tome, prezauzet je davanjem časova u slobodnom vremenu

4 Zar da se sramoti pred decom (koja o računaru mnogo više znaju)?

Pantelija neće da se meša i opet ste vi na sceni Jocu kuvate na više načina

— ističite kako ćete ostati informatički nepismeni (jecajte pri tom),

— navodite kako vam je ugrožena životna perspektiva (ovde već možete i da se ozbiljno rasplačete),

— decidirano tvrdite da ako se ovako nastavi naša zemlja nikad neće ući u Eureku (ovde slobodno lijte suze krokodilske)

Našto će upaliti Ako još i Pantelija obeća koji bod više Joci onda ste prešli u treću fazu Jer avaj, računar za koji ste se jedva izborili ne odgovara standardima Tražite pravi u fazi koja je uglavnom pucačka

Kako i ne bi kad je u sve umešen i Interpol Njegovim pouzdanim kanalima saznali ste da se cena osnovne i preporučene PC konfiguracije vrti oko 600 dolara

A kod nas? Boli glava

Doduše mi nemamo Silicijumsku dolinu, ali imamo Marićevića jarugu, odakle je Karađorđe svojedobno trešnjevim topom razgonio Turke na buljuke zamislite sad šta bi tek mogao trešnjev čip

Tražeći najmanje skupu konfiguraciju, obilazite sledeće objekte

1 Fabriku samohodnih pampura (izum prikladan za ienje pijance bez otpirača u kući)

2 Fabriku pingvina od domaćih sirovina

3 Neidentifikovanu novovaku zadužbinu za koju se pouzdano ne zna šta proizvodi

Za sve ove raznorodna lokacija bitno je (i zajedničko) samo jedno sve vam one nude domaći PC

Dok ste kupac ako ste uopšte kupac, teku med i mleko, ali kad otkrijete svoje prave namere i pokušate da zabodete nos u njihovu računicu i tzv strukturu cena, onda neka vam Svevišnji bude na ispomoć dragi naš D V i to najmanje zbog čepova i pingvina u jurišu

Uostalom naša je šansa domaća pamet i domaći softver — ali to je neka druga igra.

Miloje Gašić

RANDOMIZE GAME

Bog i šeširdžija

Kad razgovarate s nekim „iz prosvete“, po pravilu očekujete ono opšte mesto Kompjuterske igre negativno deluju na formiranje karaktere' Ili već tako nekako Slučajni poznanik nastavnik bio je možda stoti od koga sam čuo takvu tvrdnju nedavno Sit argumenata ***za i protiv,*** *nisem ništa rekao, ali biće da mi se nešto videlo na licu jer nastavnik reče — Nije ono što ste mislili Imam na umu poentiranje, obnavljanje energije i slično To kod dece stvara pogrešnu predstavu o životu*

— A meni se to čini pozitivnim — rekoh — Kad igrač uzme pravi predmet ili ukloni prepreku dakle kad uradi nešto ***korisno,*** *biva* ***nagrađen*** *dobija poene ili novu energiju, ili nešto treće Obratno kad učini nešto* ***štetno,*** *biva* ***kažnjen.*** *Zar to nije ispravno? Zar to nije vaspitno?*

— Ne — reče nastavnik — jer nema veze sa stvarnim životom

Nemojte reći na času srpskohrvatskog . . .

da je Branko Radičević naš prvi pesnik kompjuterskih igara Rezultat će biti podsmeh i/ili kec. Ali pogledajte „Đački rastanak' i videćete ***Igra beše jedino mi blago.*** *Nešto dalje, u istoj pesmi, stihovi*

Mloge li se tako kad i kada
U se glava pouzdala mlada,
Telo čilo a umešna ruka . . .

rečito svedoče o pesnikovoj igračkoj inteligenciji i veštini rukovanje džojstikom Idući još dalje, nalazimo da je volao igre platformske i skupljačke ***Seri, nosi čas dole čas gore*** *Drage su mu bile i simulacije borilačkih sportova*

Mačem biješ, mačem sečeš,
Mačem sebi blago tečeš.

Na žalost preterivao je U pesmi „Kad mildijah umreti'' upečatljivo govori o posledicama predugog igranja·

Ruka lomna, telo izmoždeno,
A klecа mi slatačko koleno!

Pesnici odista idu ispred svog vremena

Dobre vesti

Kompjuteri ulaze u škole Organizovano sistematski, na osnovu propisa Škole više neće biti one stare

Zvonce se ukida Pred početak časa na ekranima će se ispisivati START GAME a na kraju GAME OVER Umesto časova fizičkog vaspitanja, igraće se ***Match Day, Yie Ar Kung Fu, Decathlon, Winter Events*** *itd, već prema planu i programu Umesto istorije* ***Kokotoni Wilf, Gauntlet*** *i 1942 Umesto biologije* ***Monty Mole, Ranarama, Scooby Doo, Donkey Kong*** *i* ***Sabre Wulf*** *Ocene će se dobijati prema broju poena u* ***Skooldaze,*** *neće biti upisivanje u viši razred nego ulaska u viši nivo*

I neće više niko bežati sa časova Umesto bežanja, igraće ***Mikie***

Kako je kod vas?

Retke su porodice koje imaju dva televizora

U ostalima se dešavaju svakodnevne drame Deci ekran treba za igre a roditeljima za ***Dnevnik*** *i serijske filmove, neko mora da izgubi*

Negde je deci određen termin (recimo od šest do pola osam), negde se kompjuter uključuje i isključuje zavisno od TV programa za ovaj ili onaj dan, negde je, opet sve prepušteno slučaju i svađi

Kako je to rešeno u vašoj porodici? Javite ***Računarima*** *Verujemo da se može naći kako kažu političari „obostrano prihvatljiva formula Zapravo nekoliko takvih formula Ko zna koliko će vam igrača (i neigrača) biti zahvalno*

EVERGREEN/NEVERGREEN

Vladan Aleksić

Evergreen

FIST II

Prešlo je vrama igara kao što su Commando i Rambo gde čovekov život vredi koliko i dobra puška (čitaj mitraljez ili bazuka). U nastavku slavnog „Puta eksplodirajuće pesnice" vaš život zavisi isključivo od vaše snage izdržljivosti i veštine. Umesto antiruskih i antivijetnamskih pobuda vaš zadatak je da pronađete drevne trigrame na kojima su zapisane najveće tajne karatea. Ekrani se ne menjaju kao kod prethodnika već skroluju, dok se borite sa majstorima karatea, kung-fua, nindicua, tea kwondoa, čie čuen taoa i ostalih borilačkih veština. Grafika je fenomenalna, nimalo testerasta. Muzika se menja na svakom nivou. Imate na raspolaganju 21 udarac i blokadu. Borite se u šumi, u jezeru, u kući, u pećini, na poljani, u vulkanu i ostalim enterijerima i eksterijerima. Na kraju posle borbe sa zlim osvajačima očekuje vas

Nevergreen

LEGEND OF KAGE

Kada sam prvi put učitao ovu igru nisam mogao da verujem da je ispred mene slavni „komodor 64". Da nisam nazreo neke boje i čuo neko zavijanje iz zvučnika mog monitora pomislio bih da je ispred mene „galaksija" ili u najboljem slučaju ZX 81, jer je onako očajna grafika samo njih dostojna. Likovi su veoma sitni i izgledaju kao da su napravljeni grafičkim simbolima. Drveće liči na debele motke za sušenja veša. Naoružani ste srpom i šurikanima (šakenima) koje veoma teško izbacujete. Uz to možete da pravite skokove od 20—30 matara. Vaši protivnici ninja znaju ninjdicu koliko i francuski kiošari. Sasvim nerealna igra nema šta

Poukova mreža

za „komodor [illegible], [illegible]+4

EXORCIST
POKE 10000 255
FIRE ANT
POKE 7199 234 POKE 7200, 234 POKE 7201 234 POKE 7202, 234
ICICLE WORKS
POKE 4461 255
ROBIN HOOD
POKE 8808 234 POKE 9809 234
GUNSLINGER
POKE 10425 234 POKE 10426 234 POKE 10427 234
MANIC MINER
POKE 10766 255
ARENA 3000
POKE 4471 255
OUT ON A LIMB
POKE 4791 255
ZODIAC
POKE 7848 255
SHARK ATTACK
POKE 8211 255
BANDITS OF ZEBO
POKE 11100 234 POKE 11101 234 (besmrtnost)

GUNWING FALCON
POKE 11219 234 POKE [illegible] POK[illegible] 11221 234 (besmrtnost)
SCRAMBLE
POKE 14603 234 POKE 14604 234 P[illegible] 14605 234 (besmrtnost)
STRIP POKER
Učitajte program. Izbrišite ekran i otkucajte CHAR 20 0 CHR$ (27)+„T SYS 5120 SYS 5318 Videćete lepu sliku

Saško Loškovski Skopje

za „komodor 64"
— Nakon resete ukucajte POKE 2050 [illegible] LIST. Računalo isplauje SYS resetiranog programa. Ukucajte taj SYS i program je ponovo tu
— Zaštite programe u standardnom Commodoreovom bejsiku (ne vrijedi za SIMON'S BASIC i sl.) POKE 818 32 POKE 819 32. Nakon bilo kakvog prekida programa ne može se ništa uplesti (efekt poput resetiranja)

Saša Dominković, Legrad

Stigli su nam i poukovi Željka Cigenovića [illegible] deset igara) ali — nažalost nedovol[illegible] čitki. Željko možeš li ponovo bali bez [illegible]

LIGHTFORCE	POKE 10559 189	POKE 10560 5 POKE 10561, 234 (besmrtnost)
KRATER FIGHT (OLO)	POKE 7656 189 POKE 7657 0 POKE 7658 234 (neranjivost)	
BEER BELLY BURT	POKE 34418 169	POKE 34419 0 POKE 34420, 234 (neranjivost)
	POKE 27348 169	POKE 27349 x (x broj života)
ASTERIX & M C	POKE 10382 234	POKE 10383 234 POKE 10384 234 (besmrtnost)

Dragan Brcanski Pančevo

B[illegible] ROGERS	POKE 8825 36
[illegible]NA MINER	POKE 34623 44
CONGO BONGO	POKE 3442 234 ili POKE 3477 234
CRAZY KONG	POKE 30624 173
FA CON PATROL	POKE 16764 36 SYS 16670
FORT APO ALYPSE	POKE 36339 153

Jerry Mikšić Sisak

Gotovo je, gotovo!

LOAD RUNNER RESCUE

Igre nije nova ali jeste nova. Nije zato što hronološki nije a jeste zato što se kod nas veoma malo vrtela original je na disketi i nije se dao presnimiti na kasetu. No u međuvremenu su se i kod naših igrača namnožile disketne jedinice, pa je LRR postao dostupniji.

Cilj je jednostavan: simpatična curica treba da prođe kroz 46 ekrana u svakom pokupi zadati broj ključeva i oslobodi svog dragog iz zatvora.

Pred vama je tako niz logičkih zagonetki u svakovrsnim ambijentima: planina, vodopadi, zamkovi i svašta još. Nekad hodate, nakad plivate, nekd skačete

i to je ono bitno. Skakaćete često, ali samo „u nivou" ili za po jednu stepenicu naviše ili naniže. Svi su ostali skokovi pogibljeni.

Dosta ćete i pirveti ali reke i potoci ne teku uvek onako kako im nalažu prirodni zakoni: desi im se i da krenu uza zidove tvrđava. Pažljivo posmatrajte maticu, pa ćete već naći načina da vas dovede onde gde vi hoćete. Naročito otvarajte oči na 32. i 35. nivou.

Zna se, tu su i smetale. Ovog puta imaju oblik čuvara. Sa njima možete trojako. Možete im umaći jer ste brži. Možete ih preskočiti (ako ste na istoj ravni). Možete (a neki put ćete i morati) da ih namamite u neki zabačeni deo ekrana na mesto odakle ne umeju izaći.

Poseban su problem liftovi. U principu su korisni, ali postoje i oni skriveni koji vas, taman kad pomislite da ste problem rešili, povezu nekud naniže i onda hajde iz početka.

U višim nivoima igre, moraćete skakati sa liftova dok su u pokretu — ali ne sa svake visine i ne u svakom trenutku.

U osmom nivou pojaviće se pilule koje vam omogućavaju da porastete i izvedete dvostruko veći skok. Njihovo je dejstvo privremeno; dobro procenite kad ćete ih upotrebiti.

Zista će vam vredeti osećaj za perspektivu. Dobar deo scenografije prikazuje samo polovinu nečega — recimo, zamka. Druga polovina je nevidljiva, ali simetrična prvoj. Kad to shvatite, moći ćete skakati „u prazno", a ipak doskočiti na stepenik ili platformu.

Takođe, ima i nevidljivih ključeva, nezgodnih mesta gde ne možete bez mape (recimo, 41. i 44. ekran), i sakrivenih izlaza od kojih vodi samo logičko rasuđivanje.

Kad pogubite sve živote na vraćate se na početak igre nago — srećom — u onaj nivo do koga ste stigli. Dakle još jedan kompliment programeru.

Koristeći funkcionalne dirke možete na samom početku ući u bilo koji nivo — ili u

Rubrike ***Gotova je, gotova!*** *i* ***Sveže učitano*** *(nove igre) nastaju u saradnji sa emisijom Radio-Beograda ČIP i SEDAM JARIĆA koje je na programu subotom u 14 02 (I program)*

probni mod, u koma jasno vidite sva struje u rekama i potocima, pa se tako tokom igre možete pripremiti za plivačke nivoe.

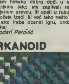
Boja su jaka i pomalo drečava, ako vam smetaju, pritisnite *C* i dobićete zelenu varijantu.

Aster *A* vraća vas na startno mesto, što će vam itekako dobro doći u kritičnim situacijama (ako ste zaplivali rekom u neželjenom pravcu, ili upali u rupu iz koje ne možete izaći, ili već tako nešto.

A ako treba oceniti igru, onda čista desetka!

Slobodan Perović

ARKANOID



Belaj počinje tako što je kosmička banka napala svemirski brod-zatvor „Arkanoid". Oslobođenim zločincima pomaže kristalna glava koja je sa sobom ostavila 32 zida. Vaša pravednička misija imala bi slabe izglede na uspeh da vam nisu u pomoć priskočili zoidi koji imaju moć da ruše zidova odbijajući se od vašeg broda.

To je okvirna priča ove igre od 33 nivoa koja treba logički proći. U prva 32 nivoa rušite zidova, a u trideset trećem borite se sa kristalnom glavom. Zidovi su sastavljeni od raznobojnih cigala, tamnoplave su neuništive, svetloplavima je potrebno 2 do 5 pogodaka, ostale je dovoljno pogoditi jednom.

Iznad zidova vrzmaju se svakojaki brodovi, njih uništavate zoidom ili, kad se spuste dovoljno nisko svojim brodom. Zoidi lete krivudavo, što vam malo komplikuje posao, ali lakše ćete ih kontrolisati uz pomoć pilula. Pilule mogu nositi ova oznake:

— C (neko vreme zadržava zoida, Trajanje zavisi od vas)

— S (loptica se usporava — ali ako uzmete još jednu dok traje dejstvo prethodne, ubrzaće se, trajanje je ograničeno)

— E (brod se širi, trajanje zavisi od vas)

— D (od jednog zoida postaju tri, izbegavajte ovu pilulu)

— B (sa strane se nakratko pojavi prolaz u sledeći nivo)

— L (kad izgubite sve živote, u novoj turi krećete od nivoa na koma ste uzeli ovu pilulu).



U trideset trećem nivou, rekosmo, borite se sa kristalnom glavom. Ona vas gađa ubistvenim pločicama. Kad je pogodite pet puta uzastopce, glava zasvetli, što znači da je vaša pobeda blizu. Još samo jedan pogodak i sve je gotovo. Kompjuter vam isplaćuje čestitku, sasvim zasluženu.

P.S. Besmrtnost možete postići i bez pouka. Na početku izaberite opciju za dva igrača. Prvim igračem trenirajte igru a drugim igrajte ozbiljno. Kad taj drugi stekne 20 000 poena, broj života će vrtoglavom brzinom porasti na 67. Svaki izgubljeni život nadoknađuje se novim poenima, a posle nekog vremena broj života se potpuno ustali, neće se smanjivati ma koliko vi „ginuli".

Vladimir Stakić

TUJAD

Evo jedne simpatične igre koja se može završiti bez poukova i bez mapa. Grafika je standardna, zvuk takođe, a animacija — naročito animacija robota — odlična.

Naravno, igra ne bi bila igra bez mnogih smetala koja ti oduzimaju energiju (neka i živote). Moraš ih ili uništiti ili izbeći da bi postigao cilj, naime sastaviti „spektrum" od delova razbacanih po svemirskoj stanici.

Najbolje je da delove skupljaš po sektorima. Čim igra počne, skreni levo. Naići ćeš na jednu od mnogobrojnih sodoba. Možeš je ubiti, ali nemoj, tada nećeš moći da prođeš kroz diskove koji se kreću gore-dole.

Oružje menjaš pritiskom na BREAK. Robota eliminišeš prvim oružjem koje dosad posle mašinke.

Kad pokupiš sve delove u datom sektoru, idu u teleport, tako će raditi sve dok ti ne ostane samo onaj teleport kojim si došao. Tada se vrati u kontrolnu sobu, kreni gore do teleporta, stani pored robota, pritisni BREAK a potom FIRE, i robot je gotov. Teleportuj se i ponovi postupak u sledećem sektoru. Kad se vratiš u kontrolnu sobu, kreni naniže, pazeći pri tom na usta koja se kreću levo-desno. Dalje idi ovako: desno, dole, desno, gore. Tada ćeš ugledati teleport.

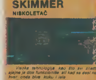
Ne zaboravi da skupljaš slika na koje naiđeš. To važi i za oružje, a energiju — samo se po sebi razume — moraš usput povremeno obnavljati. Ako ti je energija pri kraju, a nemaš je čime obnoviti, dođi do teleporta, sačekaj da energija spadne na 1, i teleportuj se, tako ćeš je popuniti do maksimuma. Kad pokupiš sve delove, vrati se u kontrolnu sobu, igra je time završena.

Srđan Stojadinović

Spektrum

SHADOW SKIMMER

NISKOLETAČ



Visoka tehnologija, kao što svi znamo, sjajna je dok funkcioniše, ali kad se desi neki kvar, onda biva kuku i lele.

Evo pogledajte. Vi ste član posade moćnog i svemoćnog vasionskog broda, došao je vaš red da rutinski prekontrolišete njegovu spoljnu površinu, sedate u svoj niskoletač, krećete na posao i zatičete ... šta?

Gomilu kvarova, ето šta. Da stvar bude još gora, to su uglavnem odbrambeni mehanizmi, a ani ne razlikuju svog od „tuđeg", dakle stavljate glavu u torbu. Povrh toga morate proći kroz sve zamke da biste se vratili odakle ste pošli. Kad vam se učini da prolaza nema, možete ili obrnuti brod za 180° ili uleteti u jedno od spremišta pa izlatiti kroz drugi otvor. U prvoj varijanti možete proći ispod nekih prepreka (ali vam je odbrambena moć umanjena), a u drugoj je dovoljno manevrisati brzo i precizno. Naravno, laserska oruđa ne haju za vaše muke, nego sve vreme ugrožavaju vaš mladi astronautski život. Čuvajte se.

WIBSTARS



[illegible]

INTO THE EAGLE'S NEST

UPAD U ORLOVO GNEZDO



Ko nije gledao u bioskopu, gledao je na TV, a ko nije ni jedno ni drugo, sad ima šansu da učita igru.

U igri će postati komandos iz II svetskog rata i ući će u osmoespratni zamak — zgradu nacističke komande — nastojeći da oslobodi tri zarobljenika, pokupi komandantovu zbirku opljačkanih antikviteta i digne zamak u vazduh.

Biće naoružan puškom, a u zamku će ovde-onde nailaziti na hranu, municiju i sandučiće za prvu pomoć. Jedna će vrata moći da razvali, a druge morati da otključava (dakle, skupljaće i ključeve).

Naravno, esesovci neće za to vreme pevati Lili Marlen, nego će nasrtati na njega, pa će biti pucnjave do mile volje (neki put i više od toga).

Pametan kao i svi komandosi, on će crtati mapu da ne bi besciljno lutao po zamku, i nastaraće se da smisli efikasan plan akcije, inače su mu slabe šanse da doživi ...

TERROR OF THE DEEP

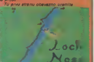
STRAH I TREPET DUBINA

Koja su to bila vremena! Šerlok Holms je rešavao slučajeve od kojih boli glava, kraljica Viktorija natucala je jezik svojih podanika, Alisa je pratila Belog Zeca po Zemlji čuda ... a u jezero Loh Nes pao je pljusak meteorita. Kasnije se ispostavilo da to nisu meteoriti nego vanzemaljska čudovišta, galaktički konkurenti omiljene Nesi.

U gnjuračkom zvonu vi se spuštate na dno jezera, prethodno odredivši mesto na mapi. Imate oružje, svetiljku, nešto vazduha, bombe, neku vrstu motora za zvono i svašta još. Prateći razne jezerske ribe, uhodite čudovišta, postavljate bombe pod njihove jazbine, udaljite se i ... BUUM!

Potom izronite po vazduh, oružje i provijant, pa jurite drugo čudovište i tako redom, sve dok Nesi ne ostane sama u jezeru ili dok vi sami ne izvisite. Na kraju vas naslov na prvoj strani novina obaveštava o tome kako ste prošli.

Tu prvu stranu obavezno uramite.



Komodor 64

SHORT CIRCUIT

KRATAK SPOJ

Spočetka ste bili robot, jedan od ko zna koliko jednakih masovnih proizvedenih ... a onda se desio neki kratak spoj i stekli ste bitne ljudske atribute: mozak i emocije. To, naravno, ne ide u račun vašem konstruktoru i njegovoj firmi, pa su se dali u poteru za vama ne bi li vas deaktivirali. U prvom činu morate pobeći na slobodu iz fabrike, a u drugom isto to, samo na otvorenom prostoru.

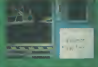
U fabrici vaš je prvi posao da istražujete sve što nađete, kao što ste radili u **Impossible Mission**. (Istraživanje, zajedno sa skakanjem, upotrebom laserskog oružja i korišćenjem nađenih predmeta, spada u dopunske funkcije. Prvo odaberite funkciju pritiskajući SPACE, a onda je kad zatreba obavljate pritiskom na FIRE.) Obratite pažnju na alarmne uređaje, i zapamtite da su mnogi predmeti beskorisni.



Posle šunjanja, pažljivog istraživanja i strpljivog skupljanja, izlazite (ako uspete) i zatičete se okruženog čuvarima. Ovde ima pucnjave do guše, i biće povuci-potegni dok ne pobegnete nekud u daljinu i belju budućnost. U beksatvu se nemojte glavom šaliti da negazite ili ičim povredite ijednu životinju. Kazna je surova i brza: umrećete od griže savesti.

Komplikovano je biti čovek — ali to nam je svima sudbina.

CHOLO



EXPLORER

ISTRAŽIVAČ

Letite vi tako po kosmosu, letite i nikome ništa zlo ne mislite, kad odjednom bum! puče nešto na vašem brodu i vi se stramoglavite na Smaragdnu planetu. Ako ne nađete devet delova koji su se razleteli i pogubili, nema vam povratka na Zemlju.

Na sreću, imate kompas, radio-lokator, devet radio farova, devet antigravitacionih „trutova", portabl mlaznicu i, naravno, laser pištolj. Možete hodati (sporo, ali sa šansom da naiđete na teleport) ili leteti (U za uzletanje, D za spuštanje), ali pre nego što ikud krenete, upotrebite radio-lokator. On će vam najčešće saopštavati da su predmeti out of range, tj. van njegovog dometa, ali će vam bar davati azimute — a i to je već nešto. Kombinujući farove i lokator moći ćete da dobijete azimut i udaljenost ... ali i to će potrajati. Planeta je velika i prevelika (40 biliona lokacija, kaže proizvođač, a vi mu verujte ako hoćete).



Nadam se da ćete, pored svih prepreka, uspeti da se vratite na Zemlju: na njoj vas, ako ništa drugo, čekaju mnoge druge uzbudljivije igre.

SUN STAR

[illegible]ČEVA ZVEZDA



Amstrad

HEAD OVER HEELS

STRMOGLAVCE



Imperija Crnozub osvojila je i podjarmila četiri planete — ali nije joj dosta. Hoće još. Planeta Sloboda — u duhu načela „bolje sprečiti nego lečiti" — rešila je da podstakne porobljene planete na ustanak i tako zavije Crnozub u crno. Ustanak će planuti tek kad budu pronađene četiri krune izgubljene u vreme kad su četiri planete izgubile rat.

Planeta Sloboda šalje uhodu — ili uhode, ako više volite. Uhoda se sastoji od dva dela — Glave i Peta — koji mogu dejstvovati samostalno, ali i spojiti se u jednu ličnost. (Otuda i naslov ***Head over heels****, što znači „strmoglavce", u doslovnom prevodu glasi „glava iznad peta".)*

Glava, očigledno, nije bila najbistrija niti su Pete blagovremeno dale petama vetar, jer Crnozupci su ih uhvatili i smestili u dve ćelije razdvojene zidom. Prvi im je problem da se nekako sastave, ako uspeju, a kad izađu, sleduje im skitnja kroz pokorene planete, skupljanje korisnih predmeta i sukobljavanje s neprijateljima. U stotom pokušaju (ili, nadam se, ranije) uspeće valjda da obave posao i trijumfalno kažu: Koskude, eto mene, eto vas, eto rata Crnozupcima.

CYBORG

KIBORG



Asteroid juri ka Zemlji kao šašav, sprema se čeoni sudar kakav još nije viđen u ovoj galaksiji, a vi ste na tom asteroidu i kolena vam se tresu.

Ubrzo nalazite svemirski brod iz kog je sedmočlana posada pobegla pred navalom vanzemaljaca. Morate naći posadu i sprečiti sudar, mada vam nije baš najjasnije kako ćete to izvesti. Lutate po brodu, mučeći nebrojene muke u borbi s karakondžulama svakojakih oblika i osobina, i nastojite da iz kompjutera u raznim prostorijama izvučete što više informacija. Pre svega, morate saznati šifre za otvaranje vrata, ali koristiće vam i podaci o čarobnim osobinama i nameni raznih tipova vanzemaljaca; na kraju je neophodno i da izađete iz programa, inače ostadoste doveka u čuvenoj GO TO petlji.

S malo sreće, naći ćete lasersku pušku, pa će vam život biti nešto lakši: moći ćete da vraćate vanzemaljcima milo za drago. No ovo nije gola pucnjava; morate mozgati, naročito kad analizirate kompjuterska izveštaje — koji su, naravno, na engleskom. Ako taj jezik ne znate, udružite se s nekim ko zna.

500CC GRAND PRIX

GRAN PRI U KATEGORIJI 500 KUBIKA

Ako ste a) amstradovac i b) skloni moto[illegible]klizmu, onda kažite: dragička vaša firma [illegible] izletela sa tri motosimulacije u isto vreme. Super Cycle i TT Racer već su bili opisani i komentarisani u ***Računarima*** *(radilo se doduše o verzijama za druge kompjutere), ova Gran Pri pojavljuje se prvi put.*

Ali više iz kurtoazije. Nećemo u njemu naći ništa istinski novo. Dvodelni ekran, izbor nivoa veštine, mogućnost igranja solistički ili udvoje — sve smo to već videli ne jednom. To ćemo videti i ovde. Neće nam biti pravo što naš vozač reaguje na naše komande malo sporije nego što je red, ali osetićemo olakšanje kad vidimo da instrument-tabla nije ni izdaleka onako utrpana i komplikovana kakva drugde ume da bude.

Od 12 ponuđenih trkačkih staza naći [illegible] svakako neku po svojoj meri — i dok [illegible] razmišljati o tome da je u [illegible] podvrstama kompjuterskih igara već odavno urađeno sve što se uraditi moglo.



WARLOCK

Vračar



Neka je promena makar i na gore. Dosad smo stalno bili na strani dobra, [illegible] postajemo vešlac, oličenje svega kvar[illegible] [illegible]skoj prirodi. Naše carstvo napale [illegible] a uz njih su (o podlosti!) i m[illegible] (verni?) duhovi, bauci i demoni. [illegible]li nije sva propalo. Na sigurnim m[illegible] [illegible]liko sigurnim da se ni mi sami [illegible]o više setiti koja su — sa[illegible] [illegible]tke koji nam daju druga obličja [illegible] postajemo snažni trol, a od d[illegible] [illegible]ljivi zloduh. Osim toga, ako na[illegible] čarobnu kuglu, naš blagočastivi p[illegible] otići će pod led dok si rekao [illegible]kus.

Naravno, tu su još ključevi, korisni [illegible] mali transporteri, razorni delovi ne[illegible] zidova itd. Sve to lepo doprinosi [illegible] ostari sazdanoj dopola od straha a [illegible] uživanja, bez koje ovakve igre ne [illegible].

Ovde imate dopunsko zadovolj[illegible] može biti loš. Ako pobedite u igri, [illegible]dost od pobede, ako izgubite, [illegible] pobedilo Dobro.